Great New Games To Run!

AUG./SEPT. 198

# ZX COMPUTING

**Britain's Biggest Magazine For The Sinclair User**




## Over 120 pages of information and programs for the ZX80, ZX81 and ZX Spectrum Computers

## PLUS

- **Business Routines**
- **DIY Keypad For ZX-81**
- **Mastering Machine Code**
- **Colourthello For Spectrum**
- **Software Reviews** — The Good, The Bad And The Ugly

# ZX COM

ZX Computing
Vol. One
Number Two
Aug/Sept 1982

**Editor:** Tim Hartnell.
**Editorial Assistant:** Kathryn McFarland.
**Advertising Manager:** Neil Johnson
**Managing Editor:** Ron Harris.
**Managing Director:** T J Connell.

Origination and design by MM Design & Print, 145 Charing Cross Road, London WC2H 0EE.

Published by Argus Specialist Publications Ltd, 145 Charing Cross Road, London WC2H 0EE.

# CONTENTS

# MPUTING

ZX Computing is published bi-monthly on the 30th of that month. Distributed by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS. 01-247 8233. Printed by: Henry Garnett Ltd., Rotherham.

Don't let its size fool you.

If anything NewBrain is like the Tardis.

It may look small on the outside, but inside there's an awful lot going on.

It's got the kind of features you'd expect from one of the really big business micros, but at a price of under £200 excluding VAT it won't give you any sleepless nights.

However, let the facts speak for themselves.

**You get what you don't pay for.**

NewBrain comes with 24K ROM and 32K RAM, most competitors expect you to make do with 16K RAM.

What's more you can expand all the way up to 2 Mbytes, a figure that wouldn't look out of place on a machine costing ten times as much.

We've also given you the choice of 256, 320, 512 and 640 x 250 screen resolution, whereas most only offer a maximum of 256 x 192.

**Big enough for your business.**

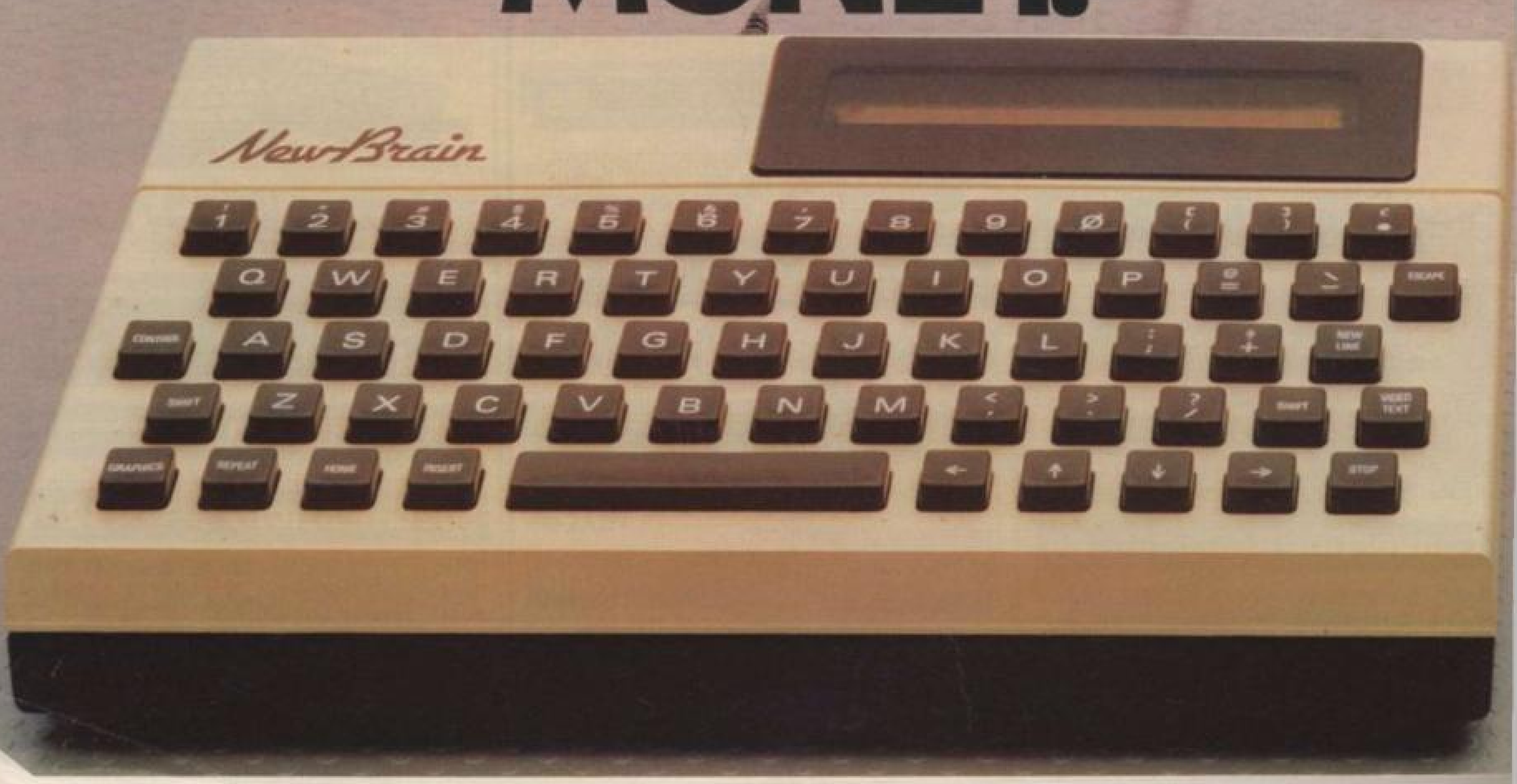
Although NewBrain is as easy as ABC to use (and child's-play to learn to use) this doesn't mean it's a toy.

Far from it.

It comes with ENHANCED ANSI BASIC, which should give you plenty to get your teeth into.

And it'll also take CP/M® so it speaks the same language as all the big business micros, and feels perfectly at home with their software.

# NO OTHER MICRO HAS THIS MUCH POWER IN THIS MUCH SIZE FOR THIS MUCH MONEY.



So as a business machine it really comes into its own.

The video allows 40 or 80 characters per line with 25 or 30 lines per page, giving a very professional 2000 or 2400 characters display in all on TV and/or monitor. And the keyboard is full-sized so even if you're all fingers and thumbs you'll still be able to get to grips with NewBrain's excellent editing capabilities.

When it comes to business graphics, things couldn't be easier. With software capabilities that can handle graphs, charts and computer drawings you'll soon be up to things that used to be strictly for the big league.

**Answers a growing need.**

Although NewBrain, with its optional onboard display, is a truly portable micro, that doesn't stop it becoming the basis of a very powerful system.

The Store Expansion Modules come in packages containing 64K, 128K, 256K or 512K of RAM. So, hook up four of the 512K modules to your machine and you've got 2 Mbytes to play with. Another feature that'll come as a surprise are the two onboard V24 interfaces.

With the aid of the multiple V24 module this allows you to run up to 32 machines at once, all on the same peripherals, saving you a fortune on extras.

The range of peripherals on offer include dot matrix and daisy wheel printers, 9," 12" and 24" monitors plus 5¼" floppy disk drives (100 Kbytes and 1 Mbyte) and 5¼" Winchester drive (6-18 Mbytes).

As we said, this isn't a toy.

**It doesn't stop here.**

Here are a couple of extras that deserve a special mention.

The first, the Battery Module, means you won't be tied to a 13 amp socket. And, even more importantly, it means you don't have to worry about mains fluctuations wreaking havoc with your programs.

The ROM buffer module gives you a freedom of another sort.

Freedom to expand in a big way. It gives you additional ROM slots, for system software upgrades such as the Z80 Assembler and COMAL, 2 additional V24 ports, analogue ports and parallel ports.

From now on the sky's the limit.

**Software that's hard to beat.**

A lot of features you'd expect to find on software are actually built into NewBrain so you don't need to worry about screen editing, maths, BASIC and graphics.

However, if you're feeling practical you can always tackle household management, statistics and educational packages. And because NewBrain isn't all work and no play, there's the usual range of mind-bending games to while away spare time.

**Waste no more time.**

To get hold of NewBrain you need go no further than the coupon at the bottom of the page.

With your order we'll include a hefty instruction manual so you'll know where to start, and a list of peripherals, expansion modules, and software so you'll know where to go next.

® CP/M is the registered trade mark of Digital Research Inc.




NewBrain, Grundy Business Systems Ltd., Grundy House, Somerset Road, Teddington TW11 8TD.

Each NewBrain order will include a FREE comprehensive user manual, a catalogue of expansion modules and peripherals, and a detailed list of available software.

Please send me the following:–

| Quantity | Item | Price per item (inc. VAT & p&p) | Total |
|---|---|---|---|
| | NewBrain A | £233.00 | |
| | NewBrain AD with onboard single line display | £267.50 | |
| | Printer | £466.00 | |
| | Monitor 12" | £142.50 | |
| | | Total £ | |

I enclose a cheque/Postal Order for £______ payable to Grundy Business Systems Reader Account NewBrain, Grundy Business Systems Ltd., Grundy House, Somerset Road, Teddington TW11 8TD.

Please debit my Access Card No:______ my Barclaycard No:______

Signature______

Name______

Address______

Postcode______ ZX/6

Registered Number 1522978
VAT Number 358661618
Please allow up to 28 days for delivery.

**NEWBRAIN**

# Welcome

Welcome to the second issue of *ZX Computing*. As you can see the magazine is going from strength to strength as ZX programmers develop their skills, and share these with us. We've brought the ZX Spectrum on board with this issue with some great programs, and also, details on how you can quickly master the tremendous sound and colour possibilities of Uncle Clive's newest computer.

As well as information on Spectrum programming, we have several big, big programs to make the most of your 16K RAM pack. These include ELEPHANT'S GRAVEYARD and the very useful TELEPHONE DIRECTORY. If you're worried about the lack of READ/DATA on the ZX81, a 1K routine in this issue will put your mind at rest.

We introduce a new section in this issue — ZX Education — which looks at the work of EZUG, the Educational ZX Users' Group, and reviews educational software. We're also reviewing a number of books, a unit which makes your ZX81 talk, and a fast-moving version of GALAXIANS. We want you to make the wisest decisions when it comes to buying software or hardware add-ons for your ZX computer, so our reviewers have been brutally honest.

Machine code. If you're like me, you keep saying to yourself "I really must get to grips with machine code someday." Our 1K machine code disassembler should help you along the way. And to maximise use of your printer, we have a tremendous routine to allow you to define your own graphics. Why not write a complete lower case alphabet? No hardware modifications are required. If you do want to get your soldering iron out, our article on adding a numeric keypad to your ZX81 should interest you.

ZX80 owners are not forgotten. We have a number of programs for you, as well as programs which will run on either the ZX80 or the ZX81, and a cassette file-handling routine for the ZX80. To make the most of the wealth of programs written in other dialects of BASIC, this issue of *ZX Computing* contains a major article explaining just how easy it is to convert other BASICs to ZX BASIC.

There's much more, including the latest news from Cambridge. So if you're reading this in a newsagent, buy it and rush home immediately. And if you're at home, miles away from your ZX81, don't read a single word more until you turn on your computer. This magazine is designed to help you make the most of your ZX computer, and will do so more effectively if you run through the routines as you come to them rather than just read about them.

Thank you to those who wrote in about issue one, and to those who sent in programs. One comment which was repeated a couple of times regarded the lack of clarity of inverse letters in ZX81 program listings. One of the decisions I made when starting this magazine was to use printer listings whenever possible, to minimise the chance of bugs. If you've waded through programs in any of the computer magazines, you'll know that it is very difficult to produce bug-free listings. However, we do sacrifice a little of the appearance to use printer listings. However, I believe — the comments of certain correspondents notwithstanding — that having program listings work is more important than having them look 'pretty'. Therefore, we'll continue to use direct printer listings whenever we can. However, if you send us programs for publication, please *do not* use inverse letters in the listing. If you think certain lines would look better in inverse when you're running the program, or other lines must be inverse to make it work, make this clear in the letter accompanying the program.

## Contributions

We're on the lookout now for good programs and articles for the next issue of *ZX Computing*. Program listings are vital, along with clear instructions on what the program does, how it works, and what the user will see when he or she runs it. Any kind of programs are useful, but we are especially interested in ones which use ZX BASIC in particularly clever ways, or in ones which contain routines which can be re-used in other programs.

All contributions we use are paid for, of course, so if you'd like to contribute to forthcoming issues and make a bit of pocket money, look through the contents of this issue, and if you can write as well, or better than our present contributors, let's hear from you.

But for now, get down to enjoying this issue with your ZX computer.

**Tim Hartnell**




## Damn the Spectrum

Dear ZX Computing,
No doubt the columns of your magazine will soon be buzzing away with news of Sinclair's new "ZX Spectrum" unveiled at the rather hap-hazardly organised Earls Court Computer Fair held in April. As a quick reminder, the Spectrum is the £125.00 colour/sound, hi-res 16K micro that, taking inflation into account, is cheaper than the ZX80!

Sinclair's total secrecy of the project may have given his creation a grand theatrical entrance, but at what price. I am not talking about the VIC-20 and Texas computers that will be wiped out overnight. That's business after all. I'm referring to the small hardware designers of add-ons for the ZX81, to bring its specifications up to that approaching the Spectrum's. These people are not in direct competition with Sinclair; they are just trying to improve his product, albeit for the reasons of personal profit.

A prime example is the case of one barren stall-holder in the ZX81 village who had just brought out a ZX81 colour graphics system for £90.00. Who will buy it, when for only £35.00 extra they can get that plus lots more in the Spectrum? The fact that there are already so many ZX81 owners means that maybe these people can sell their products, but the low price of the Spectrum means that a good percentage of ZX81 owners will buy the new micro instead of *any* add-ons for their old one.

All these independent hardware companies helped to support the ZX81. Maybe they won't now be so willing to do so with the Spectrum, even though it doesn't really need that much help.

Trevor Lawford,
Kenton, Middlesex.

● *I sympathise with your point of view to some extent, but point out that if this argument was followed to its logical conclusion we'd still be using slide rules. There would have been no pocket calculators unless Sinclair was allowed to vent the spleen of his creative genius bringing new electronic goodies into our lives. And the freedom that allows Clive to produce new products is the same one that was exercised by all those people who decided to produce products for the ZX81.*

*No-one asked them to do so, and if people decide to brave the waters of private enterprise (as, indeed, I have done with my books), they deserve all the success in the world, and not the slightest bit of protection. As you say "that's business after all." There is no such thing as "small" businessmen where the ZX81 is concerned. The whole thing is a rollicking bonanza, from which every one — producers and consumers alike — has profited. Also, I'd keep in mind that Uncle C. is not killing the ZX81. For many users, such as schools, the ZX81 still represents, to use Sinclair's own words, "the ideal introduction to computing." The ZX81 is not dead, and neither are those who have supported it.*



## Sheppey club

Dear ZX Computing,
I'm starting a ZX80/81 users' club in the Sheppey and Swale area and would like to hear from any prospective members in that area. I can be contacted on Minster 872887, or an s.a.e. to my address, 24 Baldwin Road, Minster, Sheppey, Kent, will bring details.

John Schmitt,
Sheppey.

● *Please let us hear about your local users' club. As you can see from the news section of this issue, we've mentioned all the local clubs we know about. Tell us your contact address and give us an indication of the kinds of things you do. We'll publish this and get you lots of new members. We're working closely with the National ZX Users' Club, which acts as an 'umbrella' organisation to the local clubs, and we'll make sure they publicise your activities as well.*

## Swelled heads time

Dear ZX Computing,
I have just purchased your first edition of the ZX mag. Upon buying it, I thought that it would be just the same as all of the other mags on the market. I was pleasantly surprised though, as in my point of view it is the best computer magazine ever! Unlike most other mags, the programs were faultless. No sneaky little bugs hanging about.

Reading your fab mag inspired me to write the program I have enclosed. I hope it will be considered for publication.

Lee Power,
Chadderton, Lancs.

● *Thank you very much indeed for your comments. Not all the letters we received were full of such glowing praise, and not all the programs in issue one (as a later letter shows) were completely bug-free, but we did do our best, and we're glad that most of what we did worked out so well. Regarding your program: Yes, we are interested — as I said in my introduction to this issue — in seeing as many programs and articles for subsequent editions as you care to send, and all will be considered for publication.*



## Ooooooooopppps!!!

Dear ZX Computing,
Your first issue is certainly packed full of interesting things! One article in particular caught my eye! The piece on 'Othello' on page 58. There are a few misunderstandings that need to be cleared up.

The board game 'Othello' (trademark of Peter Pan Playthings Ltd) was invented in the 1970s, not in the 1880s. It was Reversi, the Victorian game upon which Othello is based, that was invented in the 1880s.

My company Mine of Information Ltd was the first in Britain to apply computer power to the intellectual idea behind the game and I might also claim to have produced the first truly low-cost program package on general sale in Britain! A Zilog Z80 machine code program listing called 'Othello' which was published in 1978 and sold for only £2!

Mine of Information Ltd trademarked the word 'Othello' in 1979 in the class of printed matter relating to computer programs. Technically therefore your article was in breach of this trademark so I would be grateful if you would redress the balance by publishing this letter in the next issue and acknowledging my company in any future articles on this topic.

Richard Ross-Langley,
Managing Director,
Mine of Information,
St Albans.

● *Sorry 'bout that. We were under the impression that the word "Othello" was part of the English language. Shakespeare certainly thought so.*



## Do you want my program?

Dear ZX Computing,
In response to your letter in the first issue of *ZX Computing*, saying that you would like readers to send their best programs to you, I have written a golf program for the 1K ZX81. I would like to send it to you.

Please could you tell me whether or not you are interested in this sort of thing, and if I should send you my program.

Caerleon Harbinson,
Rumney, Cardiff.

● *Of course. Please send in any programs you've written that you think are good enough for publication. The information given at the start of the letters section explains the kind of thing we're looking for, and how to go about sending it in.*

## Catch that bug

Dear ZX Computing,
Frank O'Hara's "new and subtle bug" in the ZX81 ROM (rubbish characters in place of zeroes in LPRINTing small decimal numbers), arises in the routine which writes the value of a numerical expression to the Printer Buffer as a string. It can be got around very easily. Since an existing string is correctly written to the Buffer, all you need to do is LPRINT the STR$ equivalent. The following program illustrates the point:

```
10 RAND
20 LET X = INT(RND*100)/100000
30 LET X$ = STR$ X
40 LPRINT X$,X
50 GOTO 20
```

The evaluation of STR$ must be done outside the LPRINT. Compare the output of the above program with what happens if you change line 40 to LPRINT STR$ X,X — which has the same bug-bitten appearance as from LPRINT X,X.

E F Harding,
University of Cambridge,
Department of Pure Mathematics and Mathematical Statistics,
Statistical Laboratory.

● *It's good to see we are being read in the best places. Thank you for explaining a way to get around the bug. It is a pity that such convoluted methods must be found to get around problems which should never have existed in the first place.*



## Northern Ireland calling

Dear ZX Computing,
Having sat all evening with your magazine, I really must congratulate you on a marvellous issue.

I bought my ZX81 last November and became disillusioned when so-called computer experts called it "a mere toy". Thanks to your great pages I can now answer back!

By the way, do you know of any other ZX80/81 owners in Northern Ireland? It would be great to meet up and swap ideas. Keep up the good work.

Maria Savage,
11 Sandyknowes Ave.,
Newtownabbey,
Co Antrim, N. Ireland.

● *Thank you for your comments. We've printed your name and address in full, in the hope that you'll be inundated with other users who want to exchange programs and ideas with you.*

## Take that, and that

Dear ZX Computing,
During the few months in which I have owned a ZX81, I have been sampling the various computer magazines.

This week I saw your magazine *ZX Computing*, and a quick initial look led me to believe it might be the best of all the magazines I had seen so far. Unfortunately, first impressions can be misleading. Although the content ideas are admirable, the total effort has been completely spoilt for me by the shoddy printing of many of the program listings.

Several pages are too badly printed for the programs to be followed, and I am too much of a novice to work out what the characters should be. In addition, I've attempted the program on page 96. As yet I have failed to make it work. I note line 4021, there is no line 4023. Your opening comments are also interesting. How can you change lines 6000-6500 and 8000-8500? They don't exist!

Consequently, your magazine has been a disappointing buy.

M G Roe,
Hinckley, Lancashire.

● *Mea culpa. As you'll have noted from my opening comments, I determined very early in the piece that we'd use program listings direct from the printer whenever possible, just to make sure that the programs were printed without bugs. The inverse letters eluded us, so we're doing out best to keep these to a minimum. I still believe accurate listings are more important than pretty looking pages. I hope you have no problems with the programs in this issue.*

*In common with other editors of computer magazines, I often get letters claiming that program X or program Y should never have been printed because it does not run. But in all cases, I have run the program myself, then dumped it direct from the printer. There is no way it cannot run. Correct — there is one way, if the person entering the program makes a mistake. I personally ran every program in the first issue, and this issue, which is dumped from the printer, and checked the other ones out very carefully. If a program does not run first time, check the program you have in your computer against the listing. Regarding the "Horrorville" program, a splendid one indeed from N. Alexander of Margate. The program works just as it is, despite the non-existent GOTO destination. The ZX81 goes to the next available line if the line specified does not exist. Certainly it would have been neater if the line read GOTO 4030, instead of GOTO 4023, but it has exactly the same effect. The word 'change' in the introduction should have read 'add', so the line read: "You can add lines 6000 to 6500, and 8000 to 8500 to enter your own adventures."*

## Do this, and that

Dear ZX Computing,
May I congratulate you on the quality of content and presentation seen in the first issue of your magazine.

I would like to make two suggestions for future issues these being:
1. Please devote as much space as possible to the ZX Spectrum.
2. Please devote as much space as possible to educational programs that will stimulate both computer awareness and general learning in children of all ages.

P.E. Bloxham,
East Leake, Leicestershire.

● *Thanks for your comments. We intend to support the three ZX computers in accordance with the approximate percentage of ZX owners they represent, with a slight bias towards the Spectrum because the number of Spectrum owners will obviously increase dramatically throughout the rest of this year. As you can see, we've included quite a bit of Spectrum material in this issue. We have also started including educational material, and hope to expand this section in future issues. Educational programs would be most welcome.*

## It worked!

Dear ZX Computing,
I have just finished your magazine's suggested modification to extend the available memory on my ZX81.

It works very well and I thank you for its inclusion in your magazine.

I have, however, two comments to make regarding drawings on page 44.
1. The circuit diagram shows the 'fourth' NAND inputs as pins 14 and 13 in error.
2. The pictorial diagram (which incidentally does not agree with the circuit diagram, re. the above point) is shown as a top view of a 7400, not an underside view as stated.

Apart from these minor points I found the article informative, cost-effective and easy to follow.

Paul A Pitts,
Senior Development Engineer,
University of Leeds,
Audio-Visual Service.

## It Didn't Work!

Dear ZX Computing,
Thank you for publishing the article on screen POKEs for the 81 and my Squareology program. Unfortunately a few errors crept in, especially to the POKEing article, who's main errors are:-

Firstly 3½K or greater memory required for this to work.
PAGE 53
"A Simple Program", Line 10 the * should be a 0.
"Another Short Prog.", Line 30 the first / should be (.
Line 40 the > should be <.
PAGE 54 "Main Program"
Line 160 should be GOTO 120.
Line 150 the "S" should be $.
Line 190 there should be a B before the <.
Line 60 LET B$ = " "
Line 80 LET B$ = B$ + CHR$ (INT (RND*11)) + "one single space"
Line 180 the second = sign, should be 33.
PAGE 55 Missed out line 390 IF H−T>500 THEN PRINT AT 8,10;"EXCELLENT"
The Squareology program has a print positioning problem, when blacking out any taken squares. This is due to lines 118 and 486 which should read:
Line 118 LET U = X*2 + 4
Line 486 PRINT AT Y − 1; (the rest of the line is as published).

I wonder if it would be possible to photo print the proven programs (in a similar manner to the printer-readout programs) to prevent these type of errors.

I have had many reports from colleagues with ZX81 — "Excellent magazine", "First worthwhile mag worth collecting", so best of luck with this and future issues.

J.A. Enness,
Poole, Dorset.

● *Your articles were fine ones, and we're sorry we detracted from them by typographical errors. The statements you've made only prove the value of having direct printer listings. We'd be very interested in getting other articles of the quality of the first two, from you or anyone else who wishes to submit such material to be considered for publication.*

## Who wants an electric car?

Dear ZX Computing,
I am sending this letter to tell you how much I enjoyed reading the first issue of your magazine. I have not yet got a ZX81 as I am trying to sell a radio-controlled car to get the computer and 16K RAM pack. I got the magazine just to look at and I can tell you now I will be getting future issues as they come out.

Peter Craven (15),
5 Moat Hall Ave.,
Peel Green, Eccles,
Manchester M30 7LR.

● *We've printed your address in full just in case one of our readers is interested in buying your car. We hope you get a ZX81 or Spectrum very shortly. And thank you for your comments on the magazine.*





# Spectrum name under fire

The British computer manufacturers MicroAPL are angry with Clive Sinclair for naming the latest Sinclair micro the ZX Spectrum. This is because last September, MicroAPL launched a computer of their own, a fancy 16-bit machine, capable of supporting 4 megabytes of RAM, and costing around £20,000. And MicroAPL called their machine "Spectrum".

They tried to register the name, but were not allowed to do so. Rob Bittlestone, one of MicroAPL's directors, said: "We were told that the name was too general purpose to be accepted as a registered name, but now we are very concerned that confusion will arise over the two machines. Customers are already commenting on what a foolish name we chose for our product which is a bit upsetting."

Uncle Clive says he is willing to listen to suggstions from MicroAPL, who would like him to place some advertisements pointing out the difference between the two products.

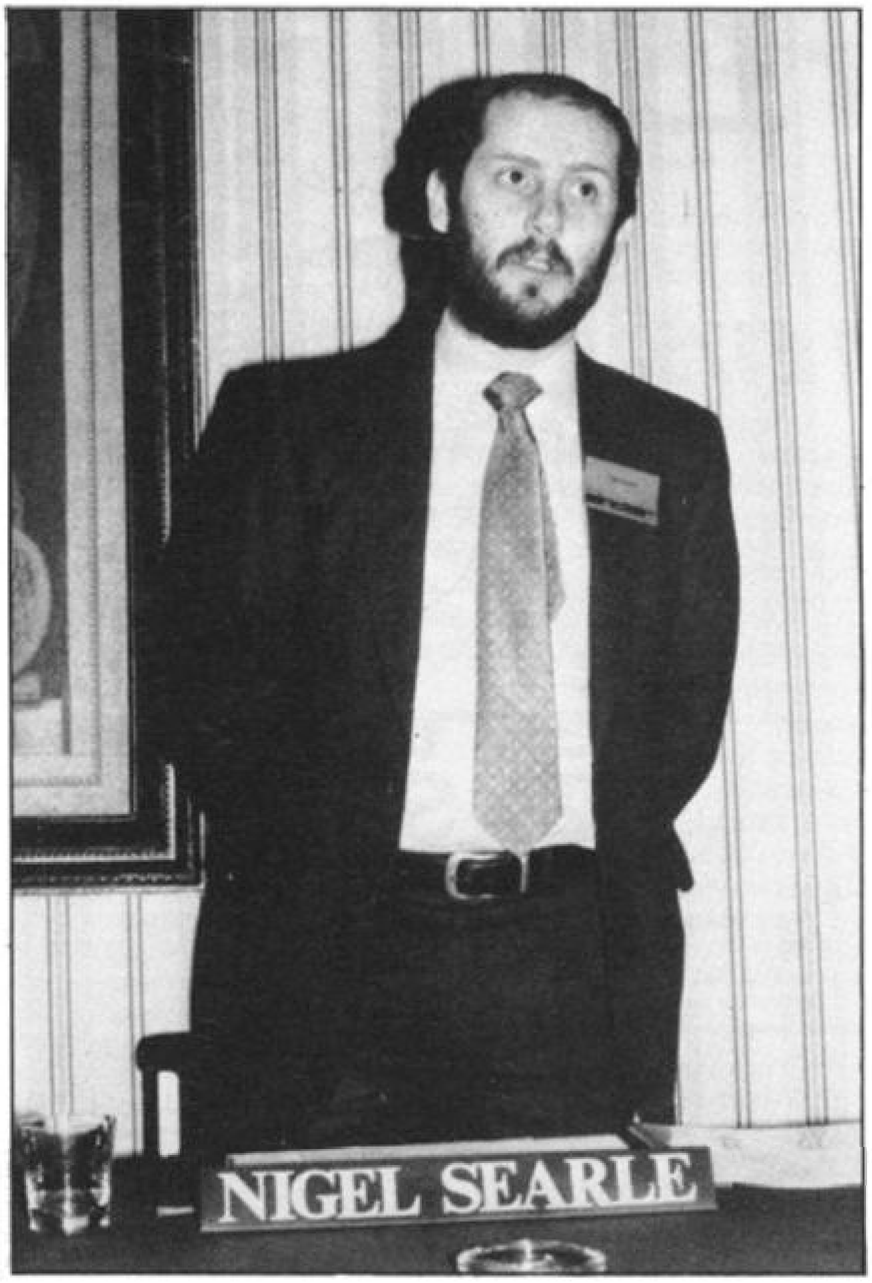
# Micro Cassette Disk

London company BATS-NCI Ltd. have announced a 'revolutionary'.new MCD-1 Micro Cassette Drive system which provides random access mass storage with compact size and high reliability. Bill Musker of BATS told *ZX Computing* that the MCD-1 "frees personal computer users from the slowness and uncertainty of saving and loading programs on tape recorders". MCD-1 is based on a small 3" single-sided floppy disk — totally enclosed with a rigid plastic cassette similar in size to ordinary cassettes.

Inside the cassette, the disk material is safeguarded against physical damage by bending, dust, scratching or greasy finger contact. Opening of the shutter, which completely covers the read/write head and drive spindle access holes, takes place automatically and only on insertion of the cassette into the front slot of the drive.

Capacity of the present version is (formatted) up to 150 Kbytes and the transfer rate is up to 250 Kbits/second. Average access times are comparable to the normal minifloppy disk. The hardware design of the drive is simple and strong and is functionally compatible with standard minifloppy controllers.

You can get more details from Bill Musker, BATS-NCI Ltd., 375b Regents Park Road, London N3 (01-349 4511/349 9217). The unit was demonstrated at the IPC Computer Faire attached to a ZX81.



*Nigel Searle.*

# New Software Launched

Clive Sinclair's sidekick Nigel Searle has announced a new range of programs for the ZX81. Many of them have been bought from Psion, and include a chess, backgammon and 'fantasy games'.

The full list of software includes:

**Cassette G3: Super Programs 3 (ICL)**
Hardware required: ZX81.
Price: £4.95.
Programs: Train Race, Challenge, Secret Message, Mind that Meteor, Character Doodle, Currency Conversion.

**Cassette G5: Super Programs 5 (ICL)**
Hardware required: ZX81 + 16K RAM.
Price: £4.95.
Programs: Martian Knock Out, Graffiti, Find the Mate, Labyrinth, Drop a Brick, Continental.

**Cassette G9: Biorhythms (ICL)**
Hardware required: ZX81 + 16K RAM.
Price: £6.95.
Programs: What are Biorhythms? Your Biorhythms.

**Cassette G10: Backgammon (Psion)**
Hardware required: ZX81 + 16K RAM.
Price: £5.95.
Programs: Backgammon, Dice.

**Cassette G11: Chess (Psion)**
Hardware required: ZX81 + 16K RAM.
Price: £6.95.
Programs: Chess, Chess Clock.
Description: Fast, efficient machine code, a graphic display of the board and pieces, plus six levels of ability.

**Cassette G12: Fantasy Games (Psion)**
Hardware required: ZX81 (or ZX80 with 8K BASIC ROM) + 16K RAM.
Price: £4.75.
Programs: Perilous Swamp, Sorcerer's Island.
Description: Perilous Swamp involves rescuing a beautiful princess from the evil wizard — with monsters lurking along the way. Sorcerer's Island is where you're marooned. To escape, and avoid the dreadful beast, you'll probably need the help of the King of Dwarfs and the Grand Sorcerer.

**Cassette G14: Flight Simulation (Psion)**
Hardware required: ZX81 + 16K RAM.
Price: £5.95.
Program: Flight Simulation (plus blank tape on side 2).

**Cassette E6: Fun to Learn series — Music 1 (ICL)**
Hardware required: ZX81 + 16K RAM.
Price: £6.95.
Programs: Composers, Musicians.

**Cassette E7: Fun to Learn *series* — Inventions 1 (ICL)**
Hardware required: ZX81 + 16K RAM.
Price: £6.95.
Programs: Inventions before 1850, Inventions since 1850.

**Cassette B1: The Collector's Pack (ICL)**
Hardware required: ZX81 + 16K RAM.
Price: £9.95.
Program: Collector's Pack, plus blank tape on side 2 for program/data storage.
Description: This comprehensive program should allow collectors (of stamps, coins etc.) to hold up to 400 records of up to 6 different items on one cassette.

**Cassette B3: VU-CALC (Psion)**
Hardware required: ZX81 + 16K RAM.
Price: £7.95.
Program: VU-CALC.
Description: Turns your ZX81 into an analysis chart. VU-CALC constructs, generates and calculates large tables for applications such as financial analysis, budget sheets, and projections.

Developed partly by ICL and partly by specialist software house, Psion, Sinclair has introduced the range in response to a widespread demand for suitable ZX81 office, educational and games software. It also believes that software increasingly holds the key to achieving continuing high sales levels.

Eight cassettes together form the 'Fun to Learn' series and are each available at £6.95, inc VAT: English Literature I and II, Geography, History, Mathematics, Inventions, Spelling and Music.

All the new cassettes require the use of the add-on 16K RAM pack with the exception of five of the ICL 'Super Programs' series, which require 1K only.

*ZX Computing* will be reviewing the new software in the next issue.

Sinclair Research have also announced changes in the prices of the 16K RAM pack and of the ZX printer. The RAM pack has dropped in price from £49.95 to £29.95, while the printer has shot up a tenner to £59.95.




*There was much beside the Spectrum to interest ZX owners at the IPC show. Here, Kayde keyboards come under scrutiny.*

# The New Computer

Journalists applauded Clive Sinclair at the end of the press conference at which he launched the ZX81. Press conferences for new machines are usually dull affairs, with journalists watching their watches and waiting for the drinks to be served, but not when Clive is centre stage.

At the IPC Computer Faire, which was the first place the Spectrum was shown publicly, the crowd was so thick around the Sinclair stand that even Clive himself gave up trying to get onto his own stand at one point, and wandered away.



*Uncle Clive exposes the ZX Spectrum to the world for the first time.*

More good news for Clive came from the Design Council who said that along with a viewdata microchip and a robot, the ZX81 deserved a council award. Hall Automation's CompArm — a robot which sprays paint — and Mullard's "Lucy" viewdata chip were the other award winners.




*You couldn't get near the Sinclair stand at the IPC Computer Faire for the crowd.*



A proliferation of computer shows — three in three weeks in London — meant thinner crowds, even though this picture of the ZX Microfair in Westminster Central Hall, taken the week after the Spectrum launch, shows that interest is still high.

# Microbrum

A major one-day exhibition/fair concentrating on the Sinclair market will take place in the centre of Birmingham on September 11th.

The venue is the Bingley Hall Exhibition Centre, a few minutes' walk from New Street Station. Microscene is set in the 8,500 square feet of Princess Hall — big enough to allow plenty of room for exhibitors and public.

One of the organisers, Eric Deeson, told us that bookings had already been made by most of the major companies in the field as well as by many of the smaller ones. Microscene hope that Sinlair Research will launch the Spectrum Micro-drive at the Birmingham show; certainly by September there will be plenty of new Spectrum software for the public to investigate.

Sinclair are, of course, increasing their support of the ZX81 in conjunction with Timex. They intend to invite Microscene exhibitors to present their wares to them at a private viewing before the public is admitted. This should be of great interest to software suppliers, particularly those hoping to make a major impact on the North American market.

Birmingham is the centre of Britain's second largest conurbation and is within easy travelling distance from the West Country, Wales, Greater Manchester and Yorkshire. Microscene are planning full advanced coverage of the show in those areas, and have already commenced negotiations for group travel with overseas organisations.

Visitors wishing to avoid the queuing problem may obtain tickets in advance at £1.00 a head. Advance copies of the Guide book also cost £1.00 each. Tickets and guide books ordered in this way will be sent out at the beginning of September.

Microscene is at 6 Battenhall Road, Harborne, Birmingham B17 9UD.

# ZX-stel

A prototype Martochoice ZX81 Prestel Adaptor was shown at the ZX Microfair, and gained considerable attention by keeping the crowd informed as to the latest events in the Falklands.

The prototype on display was designed specifically for the ZX81 in response to a competition organized by Telecom. Full two-way communications using the PRESTEL character set and attributes are used. It is not necessary to make any modifications at all to the ZX81. Although the development work has yet to be done, Martochoice say no technical difficulties are anticipated with respect to providing facilities for the use of the printer and the 16K RAM pack. This means that if you have 16K and a printer, you should be able to hold around 15 screens of Prestel.

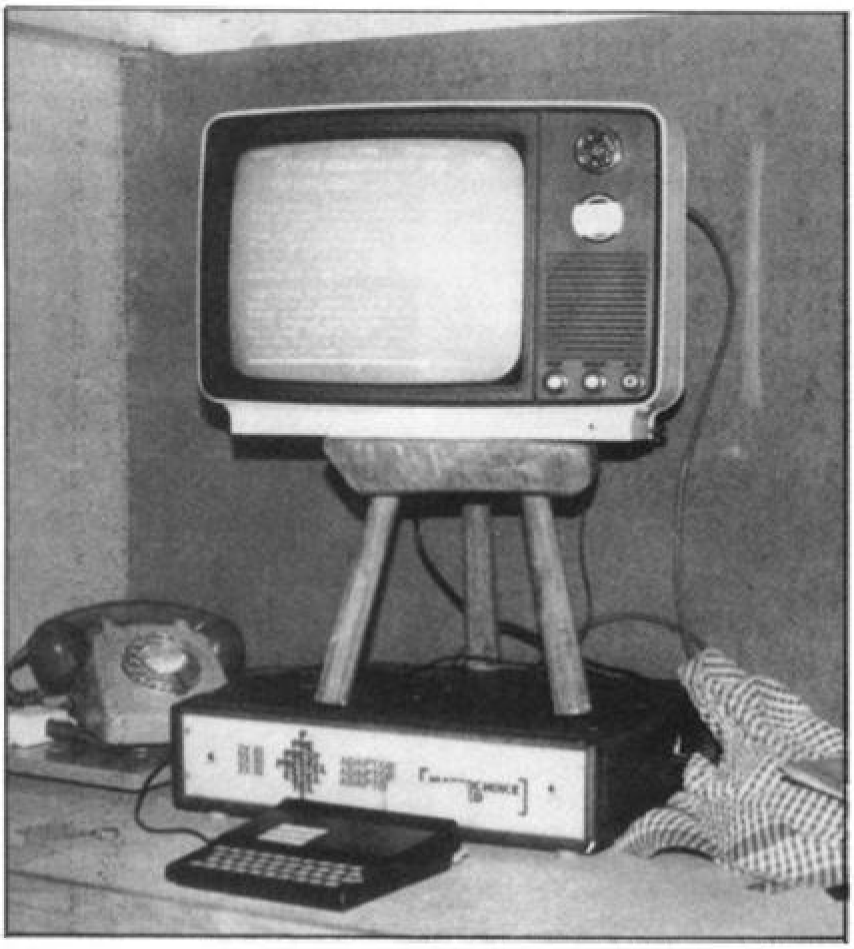
More information on the adaptor can be obtained by writing (enclosing an s.a.e.) to Martochoice Ltd., 10 Stanton Close, Jersey Farm, St Albans, Herts AL4 9HT.



ZX entrepreneur, Mike Johnson, organiser of the ZX Microfairs.



*The following week, at a ZX Microfair, the Spectrum was still the centre of attention.*



*Supported on a milking stool, a TV demonstrates Prestel, conjured up from the phone line by the ZX81 and the adaptor behind it.*

# The ZX In Belgium



*P Glennisson.*

P Glennisson has formed a ZX81 club for Flemish and Dutch people, based in Brussels. The club has been set up as a non-profit making company with social activities and is currently looking for new members.

Mr Glennisson told *ZX Computing* he is particularly interested in helping handicapped people, and one of the club's tasks is to introduce ZX81s into institutions where handicapped people live in Belgium, with the intention of teaching programming. The club publishes a 16-page A4 news letter called "ZX81 Club", and can be contacted at ZX80/81 Club, Priester de l'Epeestraat 14, B-1200, Brussels, Belgium.

# Club Roundup

The National ZX80 and ZX81 Users' Club has changed its name to the National ZX Users' Club, in response to the launch of the Spectrum. It has also decided to totally disassociate itself from the Atom and BBC activities it was involved in, and concentrate just on the ZX machines. The club's monthly magazine — INTERFACE — is now all ZX material, a development which has generally been greeted by club members. The club can be contacted (mail only) at 44-46 Earls Court Road, London W8 6EJ, and £1 will bring you a sample issue of the magazine, which features news, special offers, reviews, and many programs. Interface was the first publication in the world to publish Spectrum programs.

The ZX81 User Group, North London Hobby Computer Club, Polytechnic of North London, Holloway Road, London N7 8DB (01-607 2789), has decided to publish an occasional news sheet to be distributed among members. The first issue of the newsletter was sent to all members of the wider club, in an attempt to 'flush out' any unknown ZX owners. The newsletter also includes the following bit of information: "We've been given the business card of Chris Robins, CWR Developments, 6 Jackson Road, Islington N7 6EJ, who does ZX81 repairs. No more details, so if anyone uses him, let us know how you get on."

Jim Walsh and Paul Holmes, aided by Andrew Greening, Allan Walters, Nick Steel and John West, produce a ZX magazine called DATABUS for their school ZX81 club. They had an article on the Spectrum in an issue they were handing out a week after the new computer was launched. The rundown on the Spectrum was interesting indeed, and included the following comments from Paul Holmes:

"The Sinclair Spectrum has two principle additions to the ZX81: the ZX Microdrive capability and a full colour graphics system. It has no different modes for hi-res or text, both use the same hi-res RAMs. A number of extra commands are added, plus colour control codes. The eight colours are each labelled on the top of the keyboard and may be used in three different ways.

"PLOT provides the usual possible facilities except on a 192 x 256 grid. All 8 colours are available on the screen at once. DRAW is for drawing a line between any two points. OVER, used in conjunction with DRAW, CIRCLE, PLOT, etc. causes unCIRCLE, unPLOT, etc. FLASH, BRIGHT and INVERSE are for use with the PRINT command to achieve flashing text, two levels of brightness and inverse video. POINT is to test a hi-res point. READ, DATA, RESTORE are provided as well as multistatement lines. The sound command is BEEP and this operates the internal speaker, and has ten octaves and a single volume.

"Moving onto the ZX MICRODRIVES now: It can hold 100K bytes and takes interchangeable 3½ floppy disks. Eight drives can be connected at once. Extra commands which are provided are: CAT, producing a Disc Catalogue, ERASE for deleting a file, OPEN and CLOSE — to open and close files. The disc and tape will share a number of commands: VERIFY, MERGE, LOAD/SAVE. These are for verifications of programs, merging programs and variables and the loading and saving of programs, etc. The disk saves 16K in 3.5 seconds, whilst the cassette interface has been pushed up to 1500 Band (the ZX81 was under 300 Band). Everything is very much the same, the cursor/edit controls are basically the same.

"The Spectrum has a full ASCII character set and lower case available from the keyboard. It maintains a 24 x 32 text display, 21 user definable graphics are also available. Two control keys give TRUE VIDEO and INVERSE VIDEO, if you want to get back to normal it is quite easy.

"The Spectrum is a World beating computer, and has proved people right about Sinclair's ability, and the BBC wrong about the choice of company . . . . . WELL DONE SINCLAIR."

Other local clubs we know about include:

- EZUG (Educational ZX80/81 Users' Group), Eric Deeson, Highgate School, Birmingham 12. Send a large, stamped, addressed envelope for details. EZUG also caters for the BBC Microcomputer.
- Roger Pyatt, 23 Arundel Drive, Orpington, Kent (66) 20281.
- Austin Knott, 269 Telegraph Road, Deal, CT14 9EJ.
- Christoph Moeller, Gross Kurfurstenstrasse 41a, 4800 Bielefeld 1, Germany.
- Danmarks Nationale ZX80 og ZX81 Club, Skovmosvej 6, 4200 Slageise Dk Denmark.
- Steve Brumby, 38 Eastfield Road, Messingham, Scunthorpe, Sth. Humberside.
- Ken Knight, 22 Mount Street, Aylesbury, Bucks. HP20 2SE (0296 5181).
- David Blagden, PO Box 1 59 Kingston upon Thames, Surrey.
- Anthony Quinn, Heckenrosenweg 6, 3170 Gifhorn, W. Germany.
- Conrad Roe, 25 Cherry Tree Avenue, Walsall, WS5 4LH.
- Ian Watt, 107 Greenwood Road, Clarkeston, Glasgow.
- J. Palmer, 56 Meadowfield Drive, Edinburgh (031-661 3181).
- Leeds Microcomputer Users Group. Meets fortnightly on Thurs eve in Leeds, new members welcome. Contact: Paul O'Higgins, 20 Brudenell Mt, Leeds 6, tel: (0532) 742347 after 6.
- Brunel Computer Club: meets alternate Mondays 1900-2200 hrs at St Werburgh's Community Centre. Contact: Mr R Sampson, 4 The Coots, Stockwood.
- Worle Computer Club: meets alternate Mondays 1900-22.30 at Woodsprings Inn Function Rooms. Contact: S Rabone, 18 Castle Rd, Worle, Weston-Super-Mare, Avon, tel: 0934 513068.
- P Compton, 29 North Marine Road, Scarborough, Nth Yorks, YO12 7EY.
- Jonathan Meyer, Vanspaen Straat 22, 6524 H.N. Nymegan, Holland.
- Royston H Wallis, 22 Mallard Crescent, Pagham, Bognor Regis, West Sussex, PO21 4UU.
- Raymond Betx, Chemin du Moulin 38, 1328 Ohain, Belgium.
- Cardiff, The 81 Club. This is organised by Mike Hayes, 54 Oakley Place, Grangetown, Cardiff. Cardiff 371732.

If you'd like your club listed here, just drop a line to the National ZX Users' Club and the information will be passed on to *ZX Computing*.

# Hints 'N' Things

Thirteen-year-old James Higgo of Hertford has discovered some useful techniques to overcome common ZX problems. He listed three of them for us here at *ZX Computing*:

1. If loading fails, I pass the tape output through my Hi Fi and drop the Bass, lift the Treble and jiggle about with the various twiddly bits. This usually works after about three tries. Sometimes, however, there is an unwanted blip on the tape which cannot be eliminated. I have not used this system much as I use TDK tapes, which are usually perfect for use with my mono SANYO tape recorder. If you do not have the right sockets on your Hi Fi, you can join a few jack sockets to the leads coming from the stylus on the record player somewhere inside the Hi Fi, and also a couple — one from each speaker — for output. Most systems will have an earphone output and a microphone input anyway.
2. I have a games paddle (of a sort) on my computer which consists of five press-to-make buttons, a small box and a strip of ribbon cable. The box has the first four buttons on the top in a + formation, and one on the side for fire. The buttons are connected via ribbon cable to the computer keyboard on the underside of the PCB. The upper button to the up-arrow, the left one to the left-arrow etc, and the fire to the 0 key (actually, mine goes to 9, but 0 is more suitable in view of games like QS Defender). A joystick can be connected in place of the box and buttons. The paddle will work with most arcade games.
3. If you get stuck in a M/C routine, or want to get out of a program like ZX CHESS, I find switching a lamp off next to the computer is worth a try. The computer often gives the C error and the program is there for the saving. Sometimes a few memory locations are filled with garbage in the process and in M/C, this is hard to rectify.

# dk'tronics ZX 80/81 Hardware Software

## 4K Graphics rom



SOME OF THE GRAPHICS NOW POSSIBLE ON THE ZX81

**The dk Graphic module is our latest ZX81 accessory. This module, unlike most other accessories fits neatly inside your computer under the keyboard. The module comes ready built, fully tested and complete with a 4K graphic ROM. This will give you an unbelievable 448 extra pre-programmed graphics, your normal graphic set contains only 64. This means that you now have 512 graphics and with their inverse 1024. This now turns the 81 into a very powerful computer, with a graphic set rarely found on larger more expensive machines. In the ROM are lower case letters, bombs, bullets, rockets, tanks, a complete set of invaders graphics and that only accounts for about 50 of them, there are still about 400 left (that may give you an idea as to the scope of the new ROM). However, the module does not finish there; it also has a spare holder on the board which will accept a further 4K of ROM/RAM. This holder can be fitted with a 1K/2K/ RAM and can be used for user definable graphics so you can create your own custom character sets. £29.95.**

## ZX Keyboard

**Now with repeat key and facilities to add numeric pad. The keyboard has all the 80/81 functions on the keys. The keyboard has been specially designed for the Sinclair computer and is supplied ready-built and tested. It also has facilities for 3 extra buttons which could be used for on/off switch, reset, etc. £27.95. Numeric add on £10**

## Now from dk'tronics

a. 4K Tool Kit full of utilities to aid the programmer in constructing and debugging. E.PROM version for use with graphics ROM £9.96 cassette version £6.95.

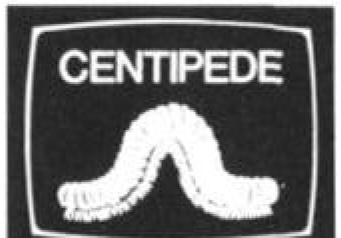
b. 16K Ram massive add on memory fully assembled and tested £19.95.

c. Flexible Ribbon Connector (as illustrated).



## 16K 81 Software

AS SEEN AT THE ZX MICROFAIR




**CENTIPEDE**
This is the first implementation of the popular arcade game on any micro anywhere. Never mind your invaders, etc., this is positively stunning, the speed at which this runs makes ZX invaders look like a game of simple snap. £4.95.




**3D/3D LABYRINTH**
You have seen 3D Labyrinth games, but this goes one stage beyond; you must manoeuvre within a cubic maze and contend with corridors which may go left/right/ up/down. Full size 3D graphical representation. £3.95.




**GRAPHIC ROM SOFTWARE**

- ★ **Asteroids**
- ★ **Space Invaders**
- ★ **Centipede**

## dk'tronics

**Buy now why wait and pay more**

**Immediate delivery**

Send off the coupon today or telephone Great Yarmouth (0493) 602453 during office hours quoting Barclaycard or Access number.
(Please add on £1 for P/P).

**23 Sussex Road,**
**Gorleston,**
**GREAT YARMOUTH,**
**Norfolk.**

Access BARCLAYCARD VISA

Please state type of machine and which ROM memory size when ordering.

Please send me ........ 4K Graphics ROM/S @ £29.95.
Please send me ........ ZX Keyboard/s @ £27.95.
(Numeric add on £10)
Please send me ........ Centipede/s @ £4.95.
Please send me ........ Labyrinth @ £3.95.
Please send me ........ Graphics ROM software Centipede @ £4.95.
Please send me ........ Space Invaders and Asteroids @ £4.95 each
Please send me ........ 4K Tool Kit @ £9.96.
(Cassette version £6.95).
Please send me ........ 16K RAM add on memory @ £19.95.
Please send me ........ Flexible Ribbon Connector/s @ £10.

I enclose £ ........

Name ........

Address ........

Please add on £1 for P/P.

Cheques/P.O. payable to DK Tronics.

# Collecting Xylenium Crystals

**From darkest Harlescott in Shrewsbury, 15-year-old Nick Wilson sends us to the planet Ganeymede 11, to gather Xylenium crystals. Monsters and matter transporters are just two of the hurdles which stand in the way of your gathering intergalactic wealth.**



In this game for a 16K ZX81 (which will run quite happily, with a few minor changes, on a ZX Spectrum), you are aboard space flight 12/03 to the planet Ganeymede 11. Your mission there is to collect as many Xylenium crystals as you can find, and bring them back to good old Terra Firma. For each crystal you bring back, you'll be paid £1,000,000.

The planet consists of an underground maze of rooms through which you can move at will. You'll get reports, during your journey, telling you about the contents of various rooms, and from time to time — if you're lucky — you'll be shown a map of the planet surface, and where you are on it. Most of the rooms are empty, but others contain things to slow you down (or even kill you). Hazards on your journey include locked rooms, monsters, matter transporters (which move you all over the place, quite randomly) and bottomless pits (and the great display which appears on your TV screen if you fall down a pit almost makes it worth ending the game in that way). Right now, gird your loins and enter the undrground caverns of Ganeymede 11.

Provision is made within the program for a SAVE of all variables, so when the program is LOADed and run it will continue from exactly where it left off. When typed in, the program should be saved by GOTO 9998, and then when loaded again it will execute itself (which sounds very painful!)

```
   1 REM *********************
   2 REM **GANEYMEDE II. NICK**
   3 REM **WILSON 82 ADAPTED **
   4 REM **FROM  GANEYMEDE BY**
   5 REM **    TIM HARTNELL  **
   6 REM *********************
   7 REM
   8 REM
   9 CLEAR
  10 LET LL=200
  15 LET CRYSTALS=0
  20 LET INIT=190
  30 LET DELAY=3000
  35 LET READ=4000
  40 LET MAP=1000
  45 DIM A(400)
  50 RAND
  60 RAND RND
  65 SLOW
  70 REM ***INSTRUCTIONS***
  75 PRINT "INSTRUCTIONS (Y/N)"
  76 IF INKEY$="" THEN GOTO 76
  77 IF INKEY$="Y" THEN GOTO 100
  78 FAST
  79 GOSUB INIT
  80 FAST
  81 GOSUB INIT
  90 SLOW
  95 GOTO 460
 100 GOSUB 9000
 130 GOTO 460
 150 FOR I=1 TO 40
 160 NEXT I
 170 CLS
 180 REM INITIALIZE
 190 FOR I=1 TO 400
 195 IF I>100 THEN FAST
 200 IF I<=20 OR I>=380 THEN LET
 A(I)=CODE "■"
 220 IF 20*INT (I/20)=I THEN LET
 A(I)=CODE "■"
 240 IF I=301 OR I=321 OR I=341
OR I=361 OR I=381 OR I=21 OR I=4
1 OR I=61 OR I=81 OR I=101 OR I=
121 OR I=141 OR I=161 OR I=181 O
R I=201 OR I=221 OR I=241 OR I=2
61 OR I=281 THEN LET A(I)=CODE "
■"
 299 IF A(I)=0 THEN LET A(I)=COD
E "."
 300 NEXT I
 301 LET INIT=305
 302 SLOW
 303 RETURN
 305 RAND
 310 FOR I=1 TO 25
 320 LET C=INT (RND*400)+1
 330 IF A(C)=CODE "." THEN LET A
(C)=CODE "C"
 355 LET C=INT (RND*400)+1
 360 IF A(C)<>CODE "■" THEN LET
A(C)=CODE "⊟"
 365 LET C=INT (RND*400)+1
 370 IF A(C)<>CODE "■" THEN LET
A(C)=CODE "P"
 375 LET C=INT (RND*400)+1
 380 IF A(C)<>CODE "■" THEN LET
A(C)=CODE "▣"
 400 NEXT I
 410 FOR I=1 TO 80
 420 LET C=INT (RND*400)+1
 430 IF A(C)<>CODE "■" THEN LET
A(C)=CODE "▒"
 440 NEXT I
 450 RETURN
 460 CLS
 470 PRINT "YOU HAVE NOW ARRIVED
 AND YOU ARENOW IN A RANDOM ROOM
....."
 475 PRINT
 476 LET OUT=INT (RND*400)+1
 477 IF A(OUT)<>CODE "." THEN GO
TO 476
 478 LET OUU=INT (RND*400)+1
 479 IF A(OUU)<>CODE "." THEN GO
TO 478
 480 PRINT "WHICH IS...";
 535 LET E=INT (RND*400)+1
 536 IF A(E)=CODE "■" OR A(E)=CO
DE "▒" THEN GOTO 535
 537 LET ORG=A(E)

 538 LET A(E)=CODE "*"
 539 PRINT E
 540 PRINT
 550 PRINT "HERE IS A MAP OF ALL
 THE ROOMS."
 560 GOSUB DELAY
 570 GOSUB MAP
 580 GOTO READ
 999 STOP
1000 CLS
1005 FAST
1010 FOR I=1 TO 400
1020 PRINT CHR$ A(I);
1030 IF 20*(INT (I/20))=I THEN P
RINT
1040 NEXT I
1045 SLOW
1050 PRINT AT 0,23;"KEY..."
1060 PRINT AT 2,22;"* -YOU"
1070 PRINT AT 4,22;". -EMPTY"
1080 PRINT AT 6,22;"▣ -MATTER"
1090 PRINT AT 7,22;"    TRANS"
1100 PRINT AT 9,22;"▒ -LOCKED"
1110 PRINT AT 10,22;"    ROOM"
1120 PRINT AT 12,22;"P -PIT"
1130 PRINT AT 14,22;"⊟ -MONSTER"
1140 PRINT AT 16,22;"C CRYSTALS"
1150 PRINT AT 18,25;"N"
1160 PRINT AT 20,25;"S"
1170 PRINT AT 19,24;"W■E"
1180 PRINT AT 21,7;"MAP FROM ABO
VE"
1200 FOR I=1 TO 100
1201 NEXT I
1202 IF LL=-50 THEN STOP
1203 CLS
1210 RETURN
3000 FOR I=1 TO 90
3001 NEXT I
3002 RETURN
3999 LET B=0
4000 CLS
4005 FAST
4010 FOR I=1 TO 32
4020 PRINT "▒";
4030 NEXT I
4040 PRINT "▒  IN ROOM NUMBER
";E;TAB (31);"▒"
4050 FOR I=1 TO 32
4060 PRINT "▒";
4070 NEXT I
4080 PRINT "▒  COMPUTER REPORTS
:";TAB (31);"▒"
4090 FOR I=1 TO 32
4100 PRINT "▒";
4110 NEXT I
4120 FOR I=1 TO 10
4130 PRINT "▒";TAB (31);"▒"
4135 NEXT I
4140 FOR I=1 TO 32
4150 PRINT "▒";
4160 NEXT I
4165 SLOW
4170 PRINT AT 6,5;
4180 IF ORG=CODE "." THEN PRINT
"EMPTY ROOM"
4190 IF ORG=CODE "C" THEN GOSUB
4500
4195 IF ORG=CODE "▣" THEN GOSUB
4700
4200 IF ORG=CODE "P" THEN GOSUB
4900
4205 IF ORG=CODE "⊟" THEN GOSUB
5000
4210 REM "WORK OUT NEXT ROOMS"
4215 LET NORTH=E-20
4216 LET EAST=E+1
4217 LET WEST=E-1
```

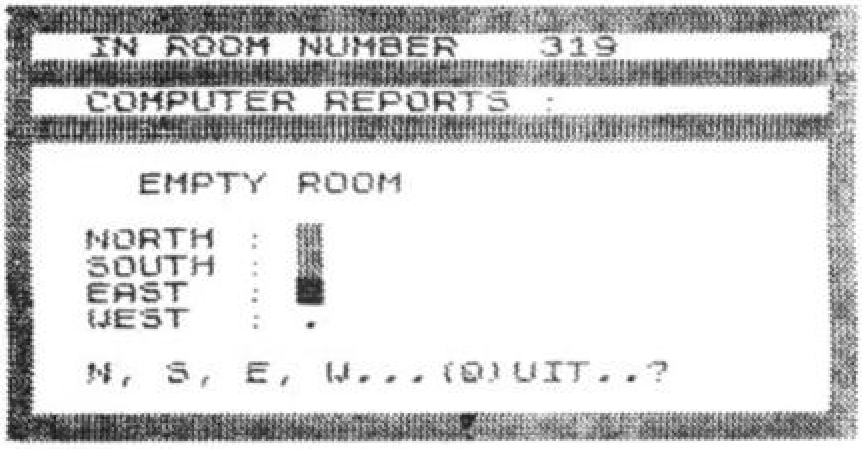
```
4218 LET SOUTH=E+20
4220 PRINT AT 8,3;"NORTH  :  ";CHR
$ A(NORTH)
4230 PRINT AT 9,3;"SOUTH  :  ";CHR
$ A(SOUTH)
4240 PRINT AT 10,3;"EAST   :  ";CH
R$ A(EAST)
4250 PRINT AT 11,3;"WEST   :  ";CH
R$ A(WEST)
4255 PRINT AT 13,3;"N, S, E, W..
.(Q)UIT..?"

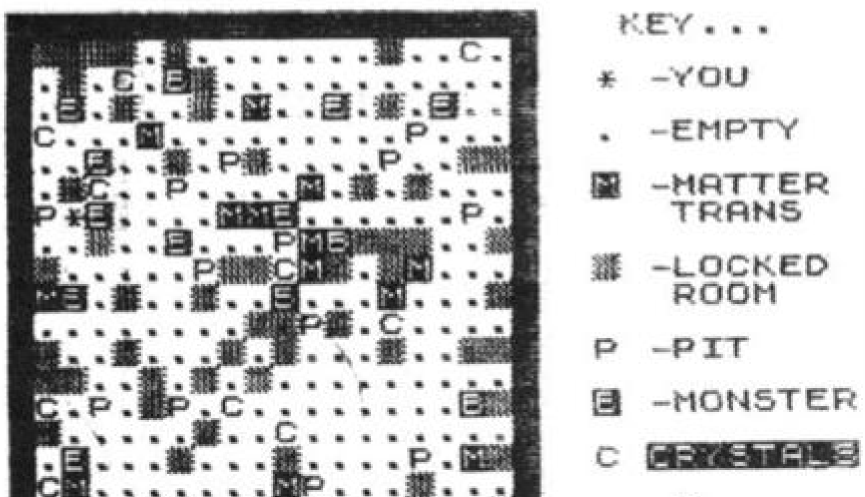
4256 LET Q1=E
4257 LET A(E)=ORG
4260 IF INKEY$="" THEN GOTO 4260
4261 IF INKEY$="Q" THEN GOTO 800
0
4262 LET M$=INKEY$
4265 IF M$="W" AND CHR$ A(WEST)<
>"■" AND CHR$ A(WEST)<>"▒" THEN
LET E=E-1
4270 IF M$="S" AND CHR$ A(SOUTH)
<>"■" AND CHR$ A(SOUTH)<>"▒" THE
N LET E=E+20
4275 IF M$="E" AND CHR$ A(EAST)<
>"■" AND CHR$ A(EAST)<>"▒" THEN
LET E=E+1
4280 IF M$="N" AND CHR$ A(NORTH)
<>"■" AND CHR$ A(NORTH)<>"▒" THE
N LET E=E-20
4285 LET ORG=A(E)
4286 LET A(E)=CODE "*"
4287 IF RND>.89 THEN GOSUB MAP
4288 IF E=OUT OR E=OUU THEN GOTO
 9600
4290 GOTO READ
4500 LET WO=INT (RND*10)+1
4505 PRINT WO;" CRYSTALS..."
4510 LET CRYSTALS=CRYSTALS+WO
4520 PRINT AT 16,3;"BRINGING YOU
R TOTAL TO ";CRYSTALS
4530 PRINT AT 17,3;30-CRYSTALS;"
 MORE TO GET."
4531 LET ORG=CODE "."
4532 LET A(E)=CODE "."
4535 RETURN
4700 PRINT "A MATTER TRANSPORTER
."
4705 LET Q1=E
4710 LET A(E)=ORG
4720 LET E=INT (RND*400)
4725 IF A(E)=CODE "■" OR A(E)=CO
DE "▒" THEN GOTO 4720
4730 PRINT AT 14,3;"YOUARE TRANS
PORTED TO ";E
4735 LET ORG=A(E)
4740 LET A(E)=CODE "*"
4745 RETURN
4900 CLS
4905 PRINT
4910 PRINT "YOU HAVE FALLEN INTO
 A PIT...."
4911 PAUSE 50
4912 LET G=RND*2
4913 LET D$="■■■■■■■■■■■■■■■■"
4920 FOR I=1 TO 21
4925 LET Y=INT (RND*7)+1
4930 SCROLL
4931 PRINT "■■■■■■■■■■■■■■■■"(Y T
O );"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"(1 TO Y*2);
"■■■■■■■■■■■■■■■■"(Y TO )
4933 IF I>4 THEN PRINT AT 17,14+
G;"▒▒▒"
4934 IF I>4 THEN PRINT AT 18,14+
G;"▒■▒"
4935 IF I>4 THEN PRINT AT 19,14+
G;"▒■▒"
4936 IF I>4 THEN PRINT AT 20,14+
G;"▒■▒"
4937 LET G=1+(RND*1)-(RND*1)
4938 NEXT I
4939 GOTO 6000
5000 CLS
5005 RAND RND
5010 LET K=INT (RND*10)+1
```

```
5020 LET ORG=CODE "."
5030 PRINT "YOU HAVE ENTERED A R
OOM WITH A  MONSTER INSIDE IT...
..."
5040 PRINT
5050 IF K<7 THEN PRINT "HE HAS S
EEN YOU....."
5055 PRINT
5060 IF K=1 THEN PRINT "AND HE E
ATS YOU UP...."
5070 IF K=1 THEN STOP
5080 IF K<7 THEN PAUSE 200
5100 IF K<4 THEN PRINT "HE IS AS
LEEP SO HE HAS NOT       NOTICED
YOU."
5110 IF K<4 THEN PAUSE 200
5120 IF K<4 THEN GOTO READ
5130 LET K=RND
5140 IF K=5 THEN LET A$="GIVES Y
OU"

5150 IF K<>5 THEN LET A$="TAKES
FROM YOU"
5160 LET FF=INT (RND*(CRYSTALS-3
))+3
5170 PRINT
5180 IF K<7 THEN PRINT "HE ";A$;
" ";FF;" CRYSTALS"
5190 IF K<7 THEN IF A$(1)="G" TH
EN LET CRYSTALS=CRYSTALS+FF
5200 IF K<7 THEN IF A$(1)="T" TH
EN LET CRYSTALS=CRYSTALS-FF
5210 PRINT
5220 IF K<7 THEN PRINT "YOUR TOT
AL IS NOW ";CRYSTALS
5231 IF K<7 THEN PAUSE 200
5240 IF K<7 THEN GOTO READ
5250 PRINT "YOU HAVE SCARED HIM
OFF...."
5260 PAUSE 200
5270 GOTO READ
6000 CLS
6005 PRINT "YOU ARE NOW OUT OF G
ANEYMEDE...."
6010 PRINT
6020 PRINT "WITH ";CRYSTALS;" CR
YSTALS...."
6025 PRINT
6030 IF CRYSTALS<30 THEN PRINT "
WHICH ARE NOT ENOUGH...."
6040 PRINT
6050 IF CRYSTALS<30 THEN PRINT "
YOU WILL HAVE TO REMAIN ON
GANEYMEDE TO DIE.... R.I.P. ..."
6060 IF CRYSTALS<30 THEN STOP
6070 PRINT "WHICH IS ENOUGH TO G
ET YOU BACK TO EARTH...."
6080 PAUSE 200
6090 CLS
6100 PRINT "YOU ARE NOW SAFELY B
ACK ON EARTHWITH ALL YOUR CRYSTA
LS INTACT."
6110 PRINT
6120 PRINT "YOU HAVE BEEN PAID
£";CRYSTALS*1E6
6130 PAUSE 300
6135 LET LL=-50
6140 GOTO MAP
8000 CLS
8005 PRINT "DO YOU WISH TO SAVE
THE DETAILS OF THESE ROOMS FOR L
ATER         CONTIUATION OF THE G
AME(Y-N)?"
8010 IF INKEY$="" THEN GOTO 8010
8020 IF INKEY$="N" THEN GOTO 600
0
8030 PRINT
8050 IF INKEY$="" THEN GOTO 8050
8060 SAVE "GANEYMEDE I■"
8065 CLS
8066 PRINT "PRESS ""E"" TO STOP
OR ""C"" TO CONTINUE."
8067 IF INKEY$="" THEN GOTO 8067
8068 IF INKEY$="E" THEN NEW
8080 GOTO READ
```

```
9000 CLS
9010 PRINT AT 0,7;"GANEYMEDE    I
I"
9020 PRINT AT 2,0;
9030 PRINT "     YOU ARE NOW ABOA
RD SPACE      FLIGHT 12/03 TO THE
PLANET        GANEYMEDE II.
                    YOUR MISSION IS
TO COLLECT  AS MANY XYLENIUM CRY
STALS AS YOUCAN FIND, AND BRING
THEM BACK TOEARTH. FOR EACH CRYS
TAL YOU DO  BRING BACK YOU WILL
BE PAID THE SUM OF £1,000,000."
9050 PRINT "     THE PLANET CONSI
STS OF AN     UNDERGROUND MAZE OF
400 ROOMS     WHICH YOU CAN MOVE A
ROUND AT      WILL.  YOU WILL HAVE
 VARIOUS      REPORTS ON YOUR JOUR
NEY, SUTCH    AS THE CONTENTS OF A
DJACENT       ROOMS, AND FROM TIME
 TO TIME A    MAP WILL BE DISPLAYE
D."
9060 PRINT
9070 PRINT "    PLEASE WAIT."
9080 GOSUB INIT
9090 CLS
9100 PRINT "      MOST OF THE ROOM
S ARE EMPTY,BUT SOME CONTAIN VAR
IOUS THINGS THAT WILL EITHER SLO
W DOWN YOUR PROGRESS, OR SPEED I
T UP.          THESE THINGS ARE
:-"
9110 PRINT
9115 PRINT
9120 PRINT "1.    ▒...A LOCKED RO
OM."
9130 PRINT
9140 PRINT "2.    ▣...A MONSTER."
9150 PRINT
9160 PRINT "3.    ▣...A MATTER TR
ANSPORTER."
9170 PRINT
9180 PRINT "4.    P...A BOTTOMLES
S PIT."
9190 PRINT
9200 PRINT "5.    C...ROOM CONTAI
NS CRYSTALS"
9210 PRINT
9220 PRINT
9230 PRINT "  PLEASE WAIT..."
9240 GOSUB INIT
9250 CLS
9260 PRINT "1.  THE LOCKED ROOM.
"
9270 PRINT
9280 PRINT "YOU CANNOT ENTER OR
PASS THROUGHA LOCKED ROOM, THEY
ARE THERE     ONLY TO SERVE AS BAR
RICADES."
9290 PRINT
9300 PRINT "PRESS NEWLINE"
9305 IF INKEY$="" THEN GOTO 9305
9310 CLS
9320 PRINT "2.  THE MATTER TRANS
PORTER."
9330 PRINT
9340 PRINT "THE MATTER TRANSPORT
ER, IF IT ISDISTURBED, WILL TRAN
SPORT YOU TOANOTHER ROOM AT RAND
OM."
9350 PRINT
9360 PRINT "PRESS NEWLINE"
9370 IF INKEY$="" THEN GOTO 9370
9380 CLS
9390 PRINT "3.   THE MONSTER."
9400 PRINT
9410 PRINT "MONSTERS CAN EITHER
BE VERY       HELPFUL OR CAN EAT Y
OU(THIS IS    RARE). THEY CAN ALSO
 GIVE OR      TAKE CRYSTALS."
9420 PRINT
9430 PRINT "PRESS NEWLINE"
9440 IF INKEY$="" THEN GOTO 9440
9450 CLS
9460 PRINT "4.    THE BOTTOMLESS
PITS."
9470 PRINT
9480 PRINT "A BOTTOMLESS PIT IS
INESCAPABLE AND SHOULD BE AVOIDE
D AT ALL      COSTS, UNLESS YOU HA
VE OVER 30    CRYSTALS."
9490 PRINT
9500 PRINT "PRESS NEWLINE"
9505 IF INKEY$="" THEN GOTO 9505
9510 CLS
9512 PRINT "      OTHER SURPRISES
ARE TWO       ROOMS WHICH, IF ENTE
RED, WILL     TRANSPORT YOU TO THE
 SURFACE."
9513 PRINT
9514 PAUSE 400
9515 PRINT
9516 PRINT "GOOD LUCK...."
9517 PAUSE 70
9518 RETURN
9600 CLS
9610 PRINT "YOU HAVE ACCIDENTLY
ENTERED A     ROOM WHICH CONTAINS
A CHUTE       THAT LEADS OUT OF GA
NEYMEDE...."
9620 PAUSE 200
9630 GOTO 6000
9997 STOP
9998 SAVE "GANYMEDE I▒"
9999 RUN
```




MAP FROM ABOVE

# Big talker

## Your ZX81 can now just about sing and dance with a number of new products on the market. Our reviewers put them through their paces.

### Keyboards

The Fuller Keyboard and case is a well designed professional keyboard for the limited ZX80/81. The extended version offers two extra keys which can be hard wired and assigned to other functions, ie. extra shift and newline keys. It also swallows the ZX completely, and holds the RAM Pack, Motherboard, power supply, and two other boards via the Motherboard. All this is held in a neat 200mm x 350mm x 60mm injection-moulded black case. Some things that I liked about the case was the "Power On" LED, and the smooth shape with no sharp corners. In the top right-hand corner of the case there are ventilation slots which stops the power supply unit (which supplies power to the ZX and Motherboard) from becoming overheated. Fuller's aim is to cut down the wires and awkward peripherals the basic ZX unit tends to attract, and this it achieves very well. I would have preferred it if the keys were stepped as on some other keyboards, but on the whole I believe the Fuller to be one of the better keyboards on the market. Fitting the ZX to the case and keyboards is quite simple as Fuller realised that most ZX users would not be "into" electronic construction.

The ZX is taken out of its own case and screwed into the top left hand corner of the Fuller case. The ZX ribbon cables are taken out and are replaced with the Fuller cables. Next, plug in the Motherboard, power supply and any RAM cards. The last operation is to stick the self-adhesive ZX functions to the key tops. With any luck, it will work.

The Fuller Keyboard and case is altogether a very valuable package for the ZX user. It speeds up data input and sorts out the layout of peripherals and leads, etc. The extended keyboard and case costs £39.95 built, £33.95 kit (plus £2.50 P&P). If you don't want a Motherboard but would rather just add a keyboard there is a standard keyboard and case available at £36.70 built, £30.70 kit. Motherboards cost £15.95 (plus 80p P&P), 16K RAM boards £35.95, and 32K RAM boards £79.95.

Details can be obtained and orders taken at: Fuller Micro Systems, The ZX Centre, Sweeting Street, Liverpool 2.





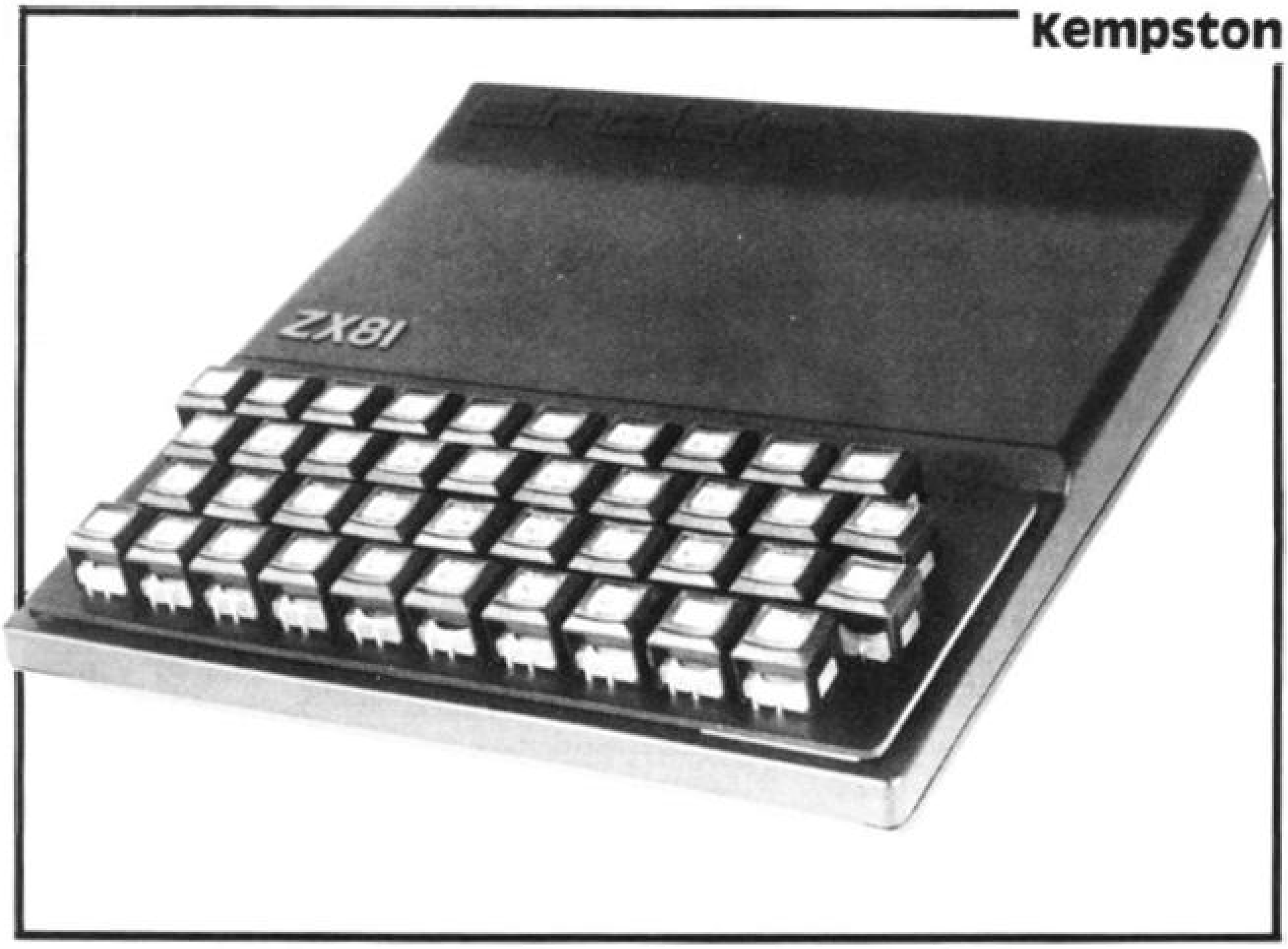
### Keen on Kayde

My first contact with Kayde Keyboards was not favourable. The first one worked erratically; the second had the "six" key upside down . . . but the third works like a dream.

The keyboard is full-size, lacking only a space bar to look and feel like a proper typewriter keyboard. One reason I chose the Kayde in the first place was because it has a repeat key, the one in the bottom left-hand corner. Although this repeats fairly slowly, it is a boon for filling a long print statement with a number of the same graphics character, or with spaces. The keyboard has increased my program entry time by about 400 per cent, and the vast majority of the programs in this issue were entered on my Kayde Keyboard.

You need to be able to solder (a little) to connect the keyboard, and I was lucky in having someone who knew how to solder to connect mine up, as I think it would have been beyond me. To connect the keyboard you remove the screws from the underside of the ZX81 and separate the two halves, then remove the two PCB securing screws and withdraw the PCB. Next you need to remove the two keyboard ribbon cables from their sockets on the PCB, and replace the PCB in its case. The wires are then connected as shown in the comprehensive assembly instructions supplied with the keyboard. The keys are blank when supplied, but a set of stick-on letters is supplied, and these are easy to apply. The transfers are on thin, tough plactic, and seem designed to withstand a great deal of wear. Even after several weeks of heavy use, my keyboard transfers show no sign of lifting off or wearing through.

All in all the keyboard is so useful I could not face the idea of going back to a ZX81 without one. The repeat key is useful for long deletions when editing, or for filling a PRINT statement with a number of the same character. The lack of quality control evident in the fact that I got two dodgey ones before a good one came along, has been pointed out to the company, who have assured me that this area of their business has been tightened up considerably. If you can solder just a little (or have a friend who you can bribe) and are sick to death of the touch-sensitive keyboard, a Kayde Keyboard will prove an asset, a boon, and a very worthwhile purchase.




## Kempston Electronics 'mini' keyboard

This tiny keyboard fits directly over the ZX81 membrane and provides a simple upgrade keyboard. Each key clicks clearly when pressed, aiding positive keying. There are no trailing wires or special cases needed. While it does not solve the problem of having the keys fairly close together, (a problem for ham-fisted typists like myself) it does provide a very good way of increasing speed of program entry, and of giving you positive feedback from each key press. The legends on the keys are identical to those on the original keys.

If you want a small, neat keyboard without the hassle of additional connector cables, the Kempston Electronics mini keyboard many be just what you're looking for. The kit is £24.50, and the fully-assembled unit £26.00 (plus 70p P&P) from: Kempston Electronics, 60 Adamson Court, Hillgrounds Road, Kempston, Beds.

## Custom Case

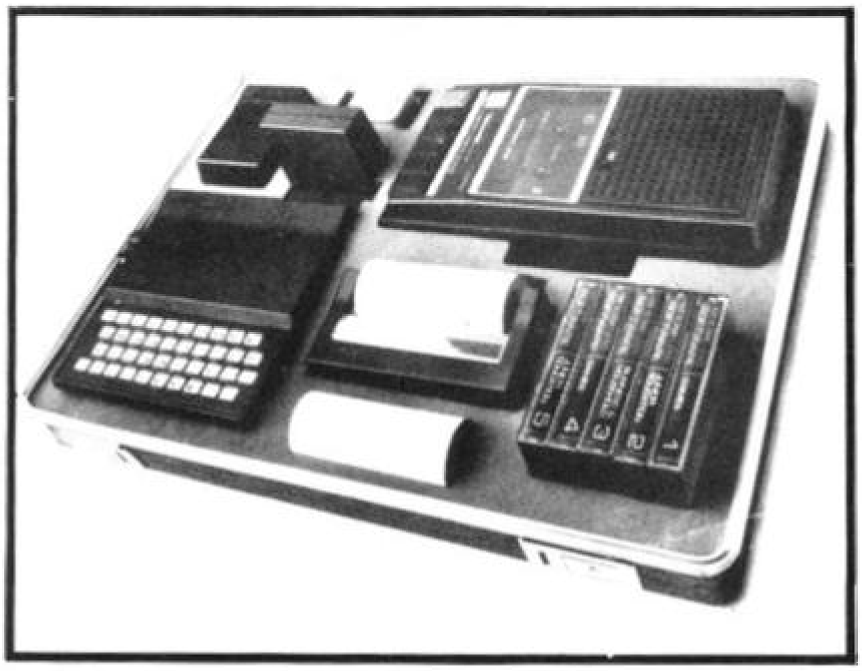
As you add extra hardware to your ZX81, it can become quite difficult keeping it all in order especially if you have to pack it away between uses. The ZX81 Custom Case (which is also available to fit a Spectrum) is designed to solve the problem. A lightweight, lockable slimline case made from some impact-resistant material ('ABS'), with foam padding, the unit is designed to hold everything firmly and safely in place.

Because the foam insert has been pre-cut to accomodate each piece of equipment, the ZX81 never has to be taken out of the case. There are no trailing connecting leads, as they all fit underneath the foam insert. Each case will hold all the standard ZX81 hardware (ie. the stuff produced by Sinclair) plus the Learning Lab and manual, software cassettes, and any cassette player up to 10½in x 5½in. If you haven't got all the hardware you can just leave the pre-cut foam where it is. This unit is ideal for those who need (and can afford) to solve their tidying-up problems in this way. It is £37.90 (plus £2.00 P&P) and is available from Computer Cases, Stanhope Road, Camberley, Surrey, GU15.

## Speech Pack

*DCP Microdevelopments' Speech Pack is easy to use, a joy to listen to, and a genuine way to enhance your programs by adding beeps (two available) and spoken word responses to your inputs. You simply connect it to the back of the ZX81 (and there is provision behind it for connecting anything else you want, like a memory pack or printer) and that's it. A single POKE command will generate a word. POKEing the specified address (49149) with zero will generate the phrase "This is Digitalker".*

*The unit (£49.95 from DCP Microdevelopments Ltd, 2 Station Close, Lingwood, Norwich, NR13 4AX) is supplied complete with Word Pack ROM 1, which contains all the letters of the alphabet, number zero to one million, and some other general words (such as: again, cent, a high tone and a low tone, and specified periods of silence). Word Pack ROM 2, available for £14.95 (as are Word Pack ROMs 3 and 4) extends its usefullness significantly, with more than 60 extra words including: minute, please, ready, start, stop, try, go, and error.*





# The Spectral Hangman rides again

## Whether you have a ZX Spectrum or a ZX81, you'll be able to run these two programs. The first, SPECTRAL HANGMAN, chooses the words you must guess. The second, TILE CRAZY, dares you to use your brain.

'Spectral Hangman' is straight forward. The computer chooses a word from its list — kept in the DATA statements in the Spectrum version, and in a series of LET lines in the ZX81 program — and then gives you a limited number of guesses (based on the length of the word) to get it right. The vocabulary for either program can easily be extended. Notice how much more compact the Spectrum version is compared with the listing for the ZX81.

The second program, 'Tile Crazy', produces a 4 x 4 square, containing the letters A to L in a random order. Your task is to put them back in alphabetical order, moving tiles into adjacent empty squares. Notice how the Spectrum version uses the INPUT option of having words within quote marks (lines 90 and 130). If you want to change the starting order, alter the two DATA lines in the Spectrum program, and A$ (see line 345) in the ZX81 version.

### SPECTRAL HANGMAN

```
 10 REM Spectral Hangman
 20 REM © K Mahogany, 1982
 30 FOR g=1 TO RND*25+1
 40 READ a$
 50 NEXT g
 60 LET n=LEN a$
 70 DIM b(n). DIM d(n)
 80 FOR g=1 TO n
 90 LET b(g)=CODE a$(g)
100 LET d(g)=b(g)
110 NEXT g
120 FOR j=1 TO n+n/3
140 GO SUB 410
190 PRINT ''' INK RND*6;"Enter
your guess no. ";j
200 INPUT c$
210 LET f=CODE c$
220 FOR g=1 TO n
230 IF d(g)=f THEN LET d(g)=0
240 NEXT g
250 NEXT j
265 GO SUB 410
270 PRINT '' INK 3;"Sorry, time
is up!"
280 GO TO 330
300 REM **** win ****
310 PRINT ' INK 4;TAB 4;"Well d
one!"
320 PRINT ' INK 4;"You got the
word in ";j-1;" guesses"
330 PRINT ' INK 2; "The word wa
s ";a$
335 PRINT '' INK RND*5;"Press a
ny key for a new game"
340 PAUSE 4e4
350 RUN
360 DATA "feature","spectrum","
cambridge","hazard","pumpkin"
370 DATA "question","quiz","fac
e","uncle","recorder"
380 DATA "basic","formula","fri
endly","resource","better"
390 DATA "butter","strawberry",
"bothersome","atom","sorcerer"
400 DATA "wizard","wickedly","e
nvy","wanton","wanderer"
410 LET h=0
415 FOR e=1 TO n
420 IF b(e)=d(e) THEN PRINT INK
RND*6;"-";
430 IF B(E)<>D(E) THEN PRINT IN
K RND*6;CHR$ (B(E));: LET h=h+1:
BEEP .25,h
440 NEXT e
445 BORDER RND*5
450 IF h=n THEN GO TO 300
460 PRINT ' INK 0; PAPER 6;"You
have guessed "; INK RND*5;h; IN
K 0;" letter";
470 IF h<>1 THEN PRINT PAPER 6;
"s"
480 PRINT
490 RETURN
```

```
Enter your guess no. 3
b-----
You have guessed 1 letter


Enter your guess no. 4
be--e-
You have guessed 3 letters
```

```
ENTER YOUR GUESS NO. 9
   10 REM SPECTRAL HANGMAN
   20 REM (C) K MAHOGANY 1982
   30 GOSUB 1000
   60 LET N=LEN A$
   70 DIM B(N)
   75 DIM D(N)
   80 FOR G=1 TO N
   90 LET B(G)=CODE A$(G)
  100 LET D(G)=B(G)
  110 NEXT G
  120 FOR J=1 TO N+N/3
  140 GOSUB 410
  150 SCROLL
  160 SCROLL
  170 SCROLL
  180 SCROLL
  190 PRINT "ENTER YOUR GUESS NO.
";J
  200 INPUT C$
  210 LET F=CODE C$
  220 FOR G=1 TO N
  230 IF D(G)=F THEN LET D(G)=0
  240 NEXT G
  260 NEXT J
  265 GOSUB 410
  270 SCROLL
  275 PRINT "SORRY, TIME IS UP"
  280 GOTO 330
  300 SCROLL
  310 PRINT TAB 8;"WELL DONE"
  315 SCROLL
  320 PRINT "YOU GOT THE WORD IN
";J-1;" GUESSES"
  325 SCROLL
  330 PRINT "THE WORD WAS ";A$
  335 SCROLL
  337 SCROLL
  340 PRINT "PRESS ANY KEY FOR A
NEW GAME"
  345 PAUSE 4E4
  350 FOR G=1 TO 24
  360 SCROLL
  370 NEXT G
  380 RUN
  410 LET H=0
  412 SCROLL
  415 FOR E=1 TO N
  420 IF B(E)=D(E) THEN PRINT "-"
;
  430 IF B(E)<>D(E) THEN PRINT CH
R$ B(E);
  435 IF B(E)<>D(E) THEN LET H=H+
1
  440 NEXT E
  450 IF H=N THEN GOTO 300
  455 SCROLL
  460 PRINT "YOU HAVE GUESSED ";H
;" LETTER";
  470 IF H<>1 THEN PRINT "S"
  480 SCROLL
  490 RETURN
 1000 LET K=INT (RND*25+1)*10+150
0
 1010 GOSUB K

 1020 RETURN
 1510 LET A$="FEATURE"
 1515 RETURN
 1520 LET A$="SPECTRUM"
 1525 RETURN
 1530 LET A$="CAMBRIDGE"
 1535 RETURN
 1540 LET A$="HAZARD"
 1545 RETURN
 1550 LET A$="PUMPKIN"
 1555 RETURN
 1560 LET A$="QUESTION"
 1565 RETURN
 1570 LET A$="QUIZ"
 1575 RETURN
 1580 LET A$="UNCLE"
 1585 RETURN
 1590 LET A$="RECORDER"
 1595 RETURN
 1600 LET A$="BASIC"
 1605 RETURN
 1610 LET A$="FORMULA"
 1615 RETURN
 1620 LET A$="FRIENDLY"
 1625 RETURN
 1630 LET A$="RESOURCE"
 1635 RETURN
 1640 LET A$="BETTER"
 1645 RETURN
 1650 LET A$="BUTTER"
 1655 RETURN
 1660 LET A$="STRAWBERRY"
 1665 RETURN
 1670 LET A$="WIZARD"
 1675 RETURN
 1680 LET A$="BOTHERSOME"
 1685 RETURN
 1690 LET A$="SORCERER"
 1695 RETURN
 1700 LET A$="ATOM"
 1705 RETURN
 1710 LET A$="WICKEDLY"
 1715 RETURN
 1720 LET A$="ENVY"
 1725 RETURN
 1730 LET A$="WANTON"
 1735 RETURN
 1740 LET A$="WANDERER"
 1745 RETURN
```

```
ENTER YOUR GUESS NO. 6
S-E----M
YOU HAVE GUESSED 3 LETTERS


ENTER YOUR GUESS NO. 7
SPE----M
YOU HAVE GUESSED 4 LETTERS


ENTER YOUR GUESS NO. 8
SPEC---M
YOU HAVE GUESSED 5 LETTERS
```

## TILE CRAZY

```
  10 REM TILE CRAZY
  20 REM (C) K MAHOGANY, 1982
  30 GOSUB 330
  40 GOSUB 200
  50 GOSUB 200
  90 PRINT AT 16,3;"WHICH ONE TO
MOVE?"
 100 INPUT X
 110 IF A(X)=CODE " " THEN GOTO
100
 120 PRINT AT 16,3;X;"          TO
WHERE? "
 130 INPUT Y
 140 IF A(Y)<>CODE " " THEN GOTO
130
 150 LET A(Y)=A(X)
 160 LET A(X)=CODE " "
 170 LET GO=GO+1
 180 GOTO 50
 200 REM *** PRINT OUT ***
 210 PRINT AT 0,3;"GO NUMBER ";G
O
 220 PRINT
 225 PRINT
 230 PRINT CHR$ A(1);CHR$ A(2);C
HR$ A(3);CHR$ A(4),"  1  2  3  4
"
 240 PRINT CHR$ A(5);CHR$ A(6);C
HR$ A(7);CHR$ A(8),"  5  6  7  8
"
 250 PRINT CHR$ A(9);CHR$ A(10);
CHR$ A(11);CHR$ A(12),"  9 10 11
```

```
12"
260 PRINT CHR$ A(13);CHR$ A(14)
;CHR$ A(15);CHR$ A(16);" 13 14 1
5 16"
320 RETURN
330 REM *** INITALISE ***
340 DIM A(16)
345 LET A$="DJNBGLAEO HMCKIF"
350 FOR B=1 TO 16
360 LET A(B)=CODE A$(B)
370 NEXT B
380 LET GO=1
410 RETURN
```

```
10 REM TILE CRAZY
20 REM © K MAHOGANY, 1982
30 GO SUB 330
40 GO SUB 200
50 GO SUB 200
90 INPUT INK 7;"Which one to m
ove? ";X
110 IF A(X)=32 THEN GO TO 90
130 INPUT INK 7;TAB 8;"To where
? ";Y
140 IF A(Y)<>32 THEN GO TO 130
150 LET A(Y)=A(X)
160 LET A(X)=32
170 LET GO=GO+1
180 GO TO 50
200 REM *** PRINT OUT ***
210 PRINT AT 0,3; PAPER 7; INK
3;"GO NUMBER "; INK 2;GO
220 PRINT : PRINT
230 PRINT INK RND*4;CHR$ A(1);C
HR$ A(2);CHR$ A(3);CHR$ A(4),"
1  2  3  4"
240 PRINT INK RND*4;CHR$ A(5);C
HR$ A(6);CHR$ A(7);CHR$ A(8),"
5  6  7  8"
250 PRINT INK RND*4;CHR$ A(9);C
HR$ A(10);CHR$ A(11);CHR$ A(12),
"    9 10 11 12"
260 PRINT INK RND*4;CHR$ A(13);
CHR$ A(14);CHR$ A(15);CHR$ A(16)
,"    13 14 15 16"
320 RETURN
330 REM *** INITIALISE ***
340 DIM A(16)
350 FOR B=1 TO 16
360 READ M
370 LET A(B)=M+64
380 NEXT B
390 LET GO=1
400 PAPER 6: BORDER 2
405 CLS
410 RETURN
420 DATA 9,14,5,2,11,6,1,4,12
430 DATA 7,-32,10,13,8,3,15
```

# Delving numerically deeper



*Dr. Frank O'Hara at home in Surrey proudly holding a ZX81 printout of the largest known prime number: 2**44497 – 1. It was discovered by Harry Nelson, 47, and David Slowinski, 25, in 1979, at the Lawrence Livermore Laboratory in California, after a two-month run on a Cray One computer. It has 13395 decimal digits. It took just over two and a half hours to calculate these digits on a ZX81, using a machine code program. The printout, which is seven feet, one and a half inches long, took 15 minutes to produce on the ZX Printer.*

**Frank O'Hara from Surbiton in Surrey helped Ian Logan decode the 8K ROM. Dr O'Hara has continued his investigations into the operation of the ZX81, and here shares with us some of his discoveries, with notes on some programs on elementary number theory for the ZX81.**

Over the past year or so, Dr O'Hara has developed a few programs on elementary number theory for the ZX81, having previously run a few such programs on a programmable calculator, a Texas TI 58, over about 2½ years from mid-1978.

A couple of these programs are "one off", ie. *ad hoc* pro grams to solve a single program. There is one of about 100 bytes of machine code which generates the decimal representation of quite large powers of 2. He used this to obtain the 13395 digits of (2 to the 44497) minus 1, the largest known prime number, discovered by Nelson and Slowinski using a Cray One computer in 1979. The program took 2 hours 31 minutes to obtain this number on the ZX81.

Another even more exciting result was given by about 400 bytes of machine code, including a multiple precision multiplication routine. This program actually proved the primality of the first 15 Mersenne primes, up to and including (2 to the 1279) minus 1, a number of 386 digits. It thus repeated some of the work of "SWAC" in 1953, going far beyond what the desk calculators had done and capturing some of the flavour of a historic moment (although 28 years later!).

Three other programs have a more general purpose flavour, I have called them:

(a) SPRF: single precision prime factorization;
(b) MPRF: multiple precision prime factorization;
(c) FE24: Fermat's theorem used to test numbers up to 24 digits long for compositeness, ie. lack of primality.

The kernel of these 3 programs is the machine code multiple precision integer division routine which finds the true integer quotient and true integer remainder of an integer of arbitrary length with respect to another arbitrarily long integer. Barden is quite mistaken when he describes this process as a "cop-out" (*How to Program Microcomputers*, by William Barden, Jr. Sams, Indianapolis, 1977, page 109). It is in fact very easy to program. It is a simple extension of the standard restoring division of one or two bytes by one byte. The shift is just a loop with its kernel as: LD A, (DE): RLA : LD (DE), A. The addition or subtraction is another loop centred on: LD A, (DE) : ADC A, (HL) : LD (DE), A; and so on. In fact the only complexity arises when one has to shorten the process in order to speed it up, as in the first of the 3 programs, SPRF. This program has a 4 byte dividend and 2 byte divisor, and needs to use the exchange resisters H', L', D' and E' to gain speed.

The first program, SPRF, finds the smallest prime factor of any odd number from 5 to 4294967255 (2 to the 32 minus 1) in not more than 20 seconds. If the number is prime, the program reports this. It goes about 1000 times as fast as the Texas calculator did. This has been achieved by a series of improvements, starting with a BASIC program that was only about 10 times as fast as the calculator. The program contains about 400 bytes of machine code; 300 or so of these are just a simple linear sieve designed to exclude multiples of 3, 5 and 7 as well as 2 and so gain a factor of 35/16 in speed. The speed has to be seen to be believed. Numbers up to 7 digits long are dealt with instantaneously. The largest 8, 9 and 10 digit primes in its range take 3, 10 and 20 seconds respectively. The program can easily be adapted to print screenfuls of results and so, for example, find the largest prime less than 2 to the 32 in one run. By using random 8, 9 or 10 digit input one can use it to see primes probably never seen before. (Only the first 10 or 11 million numbers are completely and accurately listed as prime or composite.)

The second program, MPRF, is a general purpose prime factor finder and can deal with numbers up to 77 digits long, as time permits. (The break key is active in the machine code, so one can exit from it and look at the divisor.) Its speed depends on the length of the number being examined. A 13 digit number is analysed at about 10,000 divisors a minute; a 25 digit number is analysed at about 3,000 divisors a minute; ie the divisor reaches the stated range when all odd divisors are being tried. Dr. O'Hara used this program to check the factors of the "repeated unit" numbers up to 30 ones (ie. one ninth of ((10 to the 30) minus 1)) which had already been obtained, with much more labour, using a calculator. Three of these results required the third program, too. Other uses in conjunction with the third program are mentioned below. MPRF has 182 bytes of machine code and quite a lot of BASIC to start and finish it.

The third program, FE24, tests a number N by raising 2 to the power N−1, continually reducing modulo N. If the result is not equal to 1, then N is composite. Otherwise N is called a "near-prime" or a "pseudoprime". In fact N is nearly always prime in this case, and the proof of its primality can be completed provided N−1 can be completely factorized. This involves MPRF again. Dr. O'Hara used FE24 in proving the primality of 19 ones, 23 ones and a 15 digit factor of 27 ones. He also used it to obtain results which he could not get with the calculator. He found the largest N digit prime for N up to 13. He has now extended that to 20, and is still looking at the next four. FE24 is mostly in BASIC. To raise 2 to the power M, where M is usually N−1, it first gets the binary decomposition of M. Then it repeatedly squares and, as necessary, doubles the residue, starting from 1, and reducing each time mod N. Only the reduction is in machine code, about 75 bytes. The multiple precision multiplication is in BASIC. The program is fast enough for its purpose. It tests a 24 digit number in under 20 minutes. Because of its powerful indirect method it is thus more than a million times faster in achieving its aim than MPRF on its own would be.

It would be nice to find a better technique for factorization than MPRF allows. MPRF can factorize an arbitrary 13 digit number in not more than 5 hours. To factorize larger numbers would be impractical. So far Dr. O'Hara has been lucky in that the largest run needed was about 40 minutes for a factor of about 300,000. But it would be nice if a powerful technique like the use of quadratic sieves, as described by Donald Knuth in "The Art of Computer Programming" (Addison-Wesley 1969; vol. 2, pages 345-347) could be implemented on the ZX81. Dr. O'Hara has studied this, but does not think it is feasible. The array facilities available in Z80 machine code seem to be insufficient. In any case, it may be that a 16 or 32 bit microprocessor would be needed.

A final note on what is perhaps his most spectacular result to date. The beautiful and justly famous factorization of 17 ones into the product of the two primes 2071723 and 5363222357 took under 20 minutes with MPRF, compared with 20 hours on the calculator. Of course, there is a gain of a factor of 17 in time here, since one only needs to try every 34th divisor. So one reaches 2 million in the time it would normally take to reach 120,000.

## The Assembler For SPRF

1. In order to allow plenty of room for BASIC, Dr. O'Hara started by putting RAMTOP at 96, 0 ie. at 24576d (24K; with 16K RAM attached). He then used addresses 25471 to 25913 to hold the machine code (it has many subroutine calls so is non-relocatable) and these addresses for other purposes:

26496-7: to save the contents of D'E' (not necessary, I later found).
26498-9: to save the contents of H'L' (essential to save H'L', but it could have been pushed on to the stack).
26510-1: to hold the divisor.
26512-5: to hold the number being tested.
26516-7: to hold the square root of the number being tested.
26518: to hold a flag, 1 for a prime number, 0 for a composite number.

Once the number is entered (as a string) the BASIC stores it, sets the divisor to 1 and enters the square-root ("bug-proof", because of the defective ROM). It then calls the machine code and stays there until it is ready to announce primality or print some factors. The latter will involve picking up the divisor (factor) and the quotient (cofactor) and printing up to ten digits of the quotient. So the repeated parts of testing for primality are all in machine code for speed.

2. **The Assembler**. This contains 443 bytes. The first 9 just save H'L' and D'E'. The next 351 are just a simple linear sieve which excludes multiples of 3, 5 and 7 as well as 2 and hence gains a factor of about 35/16 in speed. This part is very repetitive, and only the beginning and end are shown below. Instead of just adding 2 to the divisor (DE) it adds this sequence of numbers in a perpetual loop (after trying the values 3, 5, 7 and 11):

2, 4, 2, 4, 6, 2, 6, 4, 2, 4, 6, 6, 2, 6, 4, 2, 6, 4, 6, 8, 4, 2, 4, 2, 4, 8, 6, 4, 6, 2, 4, 6, 2, 6, 6, 4, 2, 4, 6, 2, 6, 4, 2, 4, 2, 10, 2, 10.

Each time the divisor is set, the main division subroutine is called. This does an ordinary restoring division, as explained in Zaks and Barden, with the further refinement shown in the ROM division routine, of allowing a full 32 bits (16 here) in the divisor by saving any bit of the quotient which drops into the carry. The remainder is tested for zero. If it is non-zero, the divisor is tested against the square root of the number. If it is greater, the program returns to BASIC to report a prime.

| Step | Label | Opcode | Comments |
|---|---|---|---|
| 1 | | EXX | |
| 2 | | LD (6782),HL | Save H'L'. |
| 3 | | LD (6780),DE | Save D'E'. |
| 4 | | EXX | |
| 5 | | LD DE,0003 | Set divisor to 3. |
| 6 | | CALL 64E7,DIVN | Call main division subroutine. |
| 7 | | LD DE,0005 | |
| 8 | | CALL 64E7,DIVN | |
| 9 | | LD DE,0007 | |
| 10 | | CALL 64E7,DIVN | |
| 11 | | LD DE,000B | |
| 12 | | CALL 64E7,DIVN | |
| 13 | SIEVE | INC DE | Add 2 to divisor by incrementing |
| 14 | | INC DE | DE twice. |
| 15 | | CALL 64E7,DIVN | |
| 16 | | INC DE | Add 4 to divisor by incrementing |
| 17 | | INC DE | DE 4 times. |
| 18 | | INC DE | |
| 19 | | INC DE | |
| 20 | | CALL 64E7,DIVN | |
| 21 | | INC DE | |
| 22 | | INC DE | |
| 23 | | CALL 64E7 | |
| 24 | | INC DE | |
| 25 | | INC DE | |
| 26 | | INC DE | |
| 27 | | INC DE | |
| 28 | | CALL 64E7 | |
| 29 | | LD HL,0006 | Add 6 to divisor by using HL and |
| 30 | | ADD HL,DE | the EX DE, HL instruction. |
| 31 | | EX DE,HL | Eight and ten will be added in the |
| 32 | | CALL 64E7 | same way. |

. . . sieve continues to step 205, ending with:

| Step | Label | Opcode | Comments |
|---|---|---|---|
| 201 | | LD HL,000A | Add 10 to divisor by using HL and |
| 202 | | ADD HL,DE | the EX DE,HL instruction. |
| 203 | | EX DE,HL | |
| 204 | | CALL 64E7 | |
| 205 | | JP 63A0,SIEVE | Loop for ever in the sieve. |
| 206 | DIVN | EXX | |
| 207 | | LD HL,(6790) | N is loaded into D'E'H'L'. It would |
| 208 | | LD DE,(6792) | be more efficient to do this at steps |
| 209 | | EXX | 5-8. (One day I must draw a flowchart, just for fun!) |
| 210 | | LD HL,0000 | The remainder is set to zero in HL. |
| 211 | | LD BC,20 | Initialize count to 32 decimal. |
| 212 | | AND A | Clear the carry flag. |
| 213 | DIVL | EXX | Enter the division loop. |
| 214 | | ADC HL,HL | Shift the remainder-dividend- |
| 215 | | EX DE,HL | quotient left in HLD'E'H'L'. |

```
         ADC HL,HL
         EX DE,HL
         EXX
         ADC HL,HL
         JR C,650A,SAVE    If a bit drops into the carry, go and
                           retrieve it for the quotient.
         SBC HL,DE         Trial subtract the divisor.
         JR NC,650D, NRST  Go, if no carry, to no restore.
         ADD HL,DE         Add back the divisor if there was
                           carry.
         AND A             Clear the carry and go with no bit
         JR 650E,CONT      for the quotient.
    SAVE AND A             Force no restore and one for the
         SBC HL,DE         quotient here.
    NRST SCF               Set the carry flag: one for the
                           quotient.
    CONT DJNZ 64F6,DIVL    Lop back for each bit of dividend
                           (32 times).
         EXX
         ADC HL,HL         Move last bit into quotient.
         EX DE,HL
         ADC HL,HL
         EX DE,HL
         EXX
         LD A,H            Now test the remainder.
         OR L
         JR Z,6527,FACT    Go if it is zero.
         AND A             Clear the carry.
         LD HL,(6794)      Put square root of N into HL.
         SBC HL,DE         Subtract divisor.
         RET NC            Return to sieve if more to do.
         LD A,01           Otherwise, set flag for a prime
         JR 652C,EXIT      and go to EXIT.
    FACT XOR A             Reset flag for a factor.
         LD (678E),DE      Save factor for BASIC.
    EXIT LD (6796),A       Save flag for BASIC.
         EXX
         LD HL,(6782)      Restore H'L'.
         LD DE,(6780)      Restore D'E'.
         EXX
         POP HL            Discard sieve return address.
         RET               Return to BASIC.
```

```
   1 REM "SPRF"
   2 REM OR 6▙? GOSUB ?▩? OR )▀
LN SCROLL ?)▌ LN SCROLL ?)▛ LN S
CROLL ?)" LN SCROLL ?<<LN SCROLL
 ?<<<<LN SCROLL ?<<LN SCROLL ?<<
<<LN SCROLL ?5▞ ; FOR LN SCROLL
?<<LN SCROLL ?5▞ ; FOR LN SCROLL
 ?<<<<LN SCROLL ?<<LN SCROLL ?<<
<<LN SCROLL ?5▞
   3 REM ; FOR LN SCROLL ?5▞ ; F
OR LN SCROLL ?<<LN SCROLL ?5▞ ;
FOR LN SCROLL ?<<<<LN SCROLL ?<<
LN SCROLL ?5▞ ; FOR LN SCROLL ?<
<<<LN SCROLL ?5▞ ; FOR LN SCROLL
 ?5▒ ; FOR LN SCROLL ?<<<<LN SCR
OLL ?<<LN SCROLL ?<<<<LN SCROLL
?<<LN SCROLL ?<<<<LN SCROLL ?
   4 REM 5▒ ; FOR LN SCROLL ?5▞
; FOR LN SCROLL ?<<<<LN SCROLL ?
5▞ ; FOR LN SCROLL ?<<LN SCROLL
?<<<<LN SCROLL ?5▞ ; FOR LN SCRO
LL ?<<LN SCROLL ?5▞ ; FOR LN SCR
OLL ?5▞ ; FOR LN SCROLL ?<<<<LN
SCROLL ?<<LN SCROLL ?<<<<LN SCRO
LL ?5▞ ; FOR LN SCROLL ?<
   5 REM <LN SCROLL ?5▞ ; FOR LN
 SCROLL ?<<<<LN SCROLL ?<<LN SCR
OLL ?<<<<LN SCROLL ?<<LN SCROLL
?5▒ ; FOR LN SCROLL ?<<LN SCROLL
 ?5▒ ; FOR LN SCROLL ??4? OR E(?
 GOSUB ?>? OR 5 ▞4B OR GOSUB ?
FOR GOSUB ? FOR OR GOSUB ?5▒ GOS
UB ?K▛;B/▗B GOSUB ?R(
   6 REM NEW OR GOSUB ? FOR GOSU
B ? FOR OR ?PC"EEB? GOSUB ?SQR Y
▘/▌J GOSUB ?▪?M▬? OR E▙? GOSUB ?
▩? OR LPRINT TAN
```

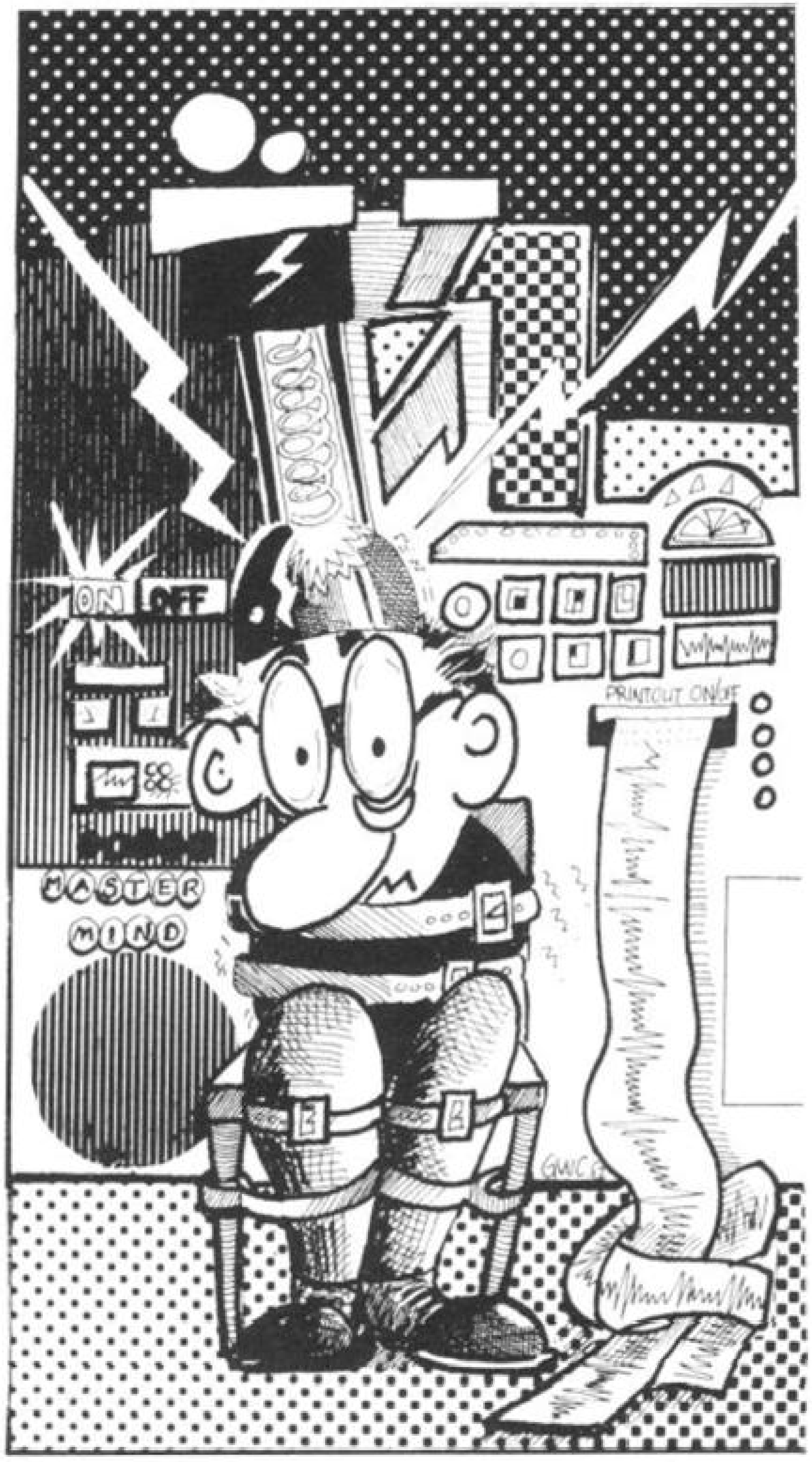
```
  10 REM ** COMMENTS AT LINE 600
  12 FAST
 120 REM *PUT CODE AT L TO L+N-1
 130 LET C=16420
 132 LET L=25371
 134 LET N=100
 140 FOR I=1 TO 5
 142 LET C=C+106
 144 LET L=L+100
 150 FOR A=L TO L+N-1
 160 POKE A,PEEK (A-L+C)
 190 NEXT A
 195 NEXT I
 315 REM **RE-ENTER HERE
 320 INPUT N$
 325 LET N=VAL N$
 330 LET H=N
 340 FOR I=0 TO 3
 350 POKE 26512+I,H-256*INT (H/2
56)
 360 LET H=INT (H/256)
 370 NEXT I
 380 POKE 26510,1
 390 POKE 26511,0
 430 LET S=10*SQR (N/100)-1
 440 POKE 26517,INT (S/256)
 450 POKE 26516,S-256*PEEK 26517
 460 LET U=USR 25471
 470 IF PEEK 26518=1 THEN GOTO 5
10
 480 LET D=PEEK 26510+256*PEEK 2
6511
 482 IF D=1 THEN GOTO 510
 485 LET Q=N/D
 490 LET Q$=STR$ INT (Q/100)
 495 IF Q-100*INT (Q/100)<10 THE
N LET Q$=Q$+"0"
 500 LET Q$=Q$+STR$ (Q-100*INT (
Q/100))
 505 PRINT N$;" = ";D;" * ";Q$
 508 GOTO 320
 510 PRINT N$;" IS PRIME"
 520 GOTO 320
 530 REM 1
 540 REM 2
 550 REM 3
 560 REM 4
 570 REM 5
 580 REM 6
 600 REM SPRF: SINGLE PRECISION
PRIME FACTORIZATION
 605 REM BY FRANK O"HARA
 610 REM FOR ODD NUMBERS FROM 3
TO 4294967295
 620 REM IDENTIFIES PRIMES OR
GIVES SMALLEST PRIME FACTOR AND
CO-FACTOR
 630 REM RUN BY KEYWORD RUN
 640 REM AFTER 7 SECS "L" WILL
APPEAR: ENTER A NO. AND NEWLINE
 650 REM CONTINUE AT WILL
 660 REM BREAK BY STOP"
 670 REM IE RUBOUT FIRST " AND
ENTER KEYWORD STOP (SHIFTED A)
 675 REM TO SEE THE SPEED, TRY:-
 680 REM "997"; "9973"; "99991";
"999983"; "9999991"; "99999989";
"999999937"; "4294967291"
 690 REM THE 1ST 5 ARE IMMEDIATE
 700 REM THE LAST 3 TAKE 3, 10
AND 20 SECONDS
 710 REM TRY "961"; "99400891";
"4294049777" TO SEE SOME FACTORS
 720 REM TO GET ALL THE FACTORS
OF A NUMBER USE Q$ AS INPUT
 730 REM EG ENTER "3202011551"
 740 REM RESULT: 1297 * 2468783
 750 REM RUBOUT THE QUOTES FROM
"L" AND ENTER Q$ (IE THE LETTER
Q AND THE SYMBOL $)
 760 REM THE RESULT: 1523 * 1621
 770 REM TRY "4294967295" (IT
HAS 5 FACTORS)
 780 REM PRIME FINDING
 790 REM ADD THESE 11 LINES:-
```

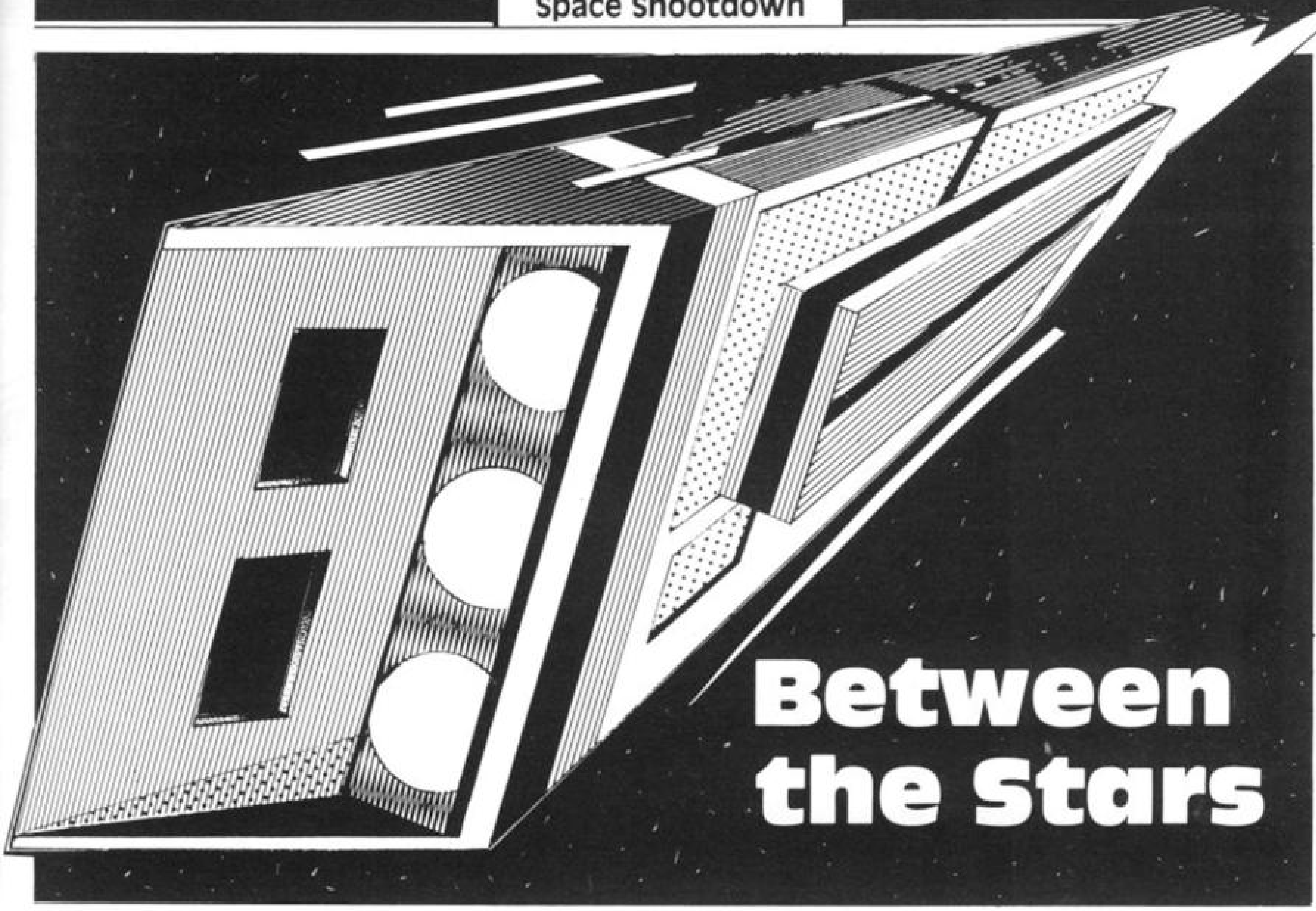
```
795 REM 321 CLS
800 REM 322 LET N=VAL N$
810 REM 325 FOR J=1 TO 20
820 REM 326 LET N=N+2
830 REM 327 LET N$=STR$ INT (N/
100)
840 REM 328 IF N-100*INT (N/100
)<10 THEN LET N$=N$+"0"
850 REM 329 LET N$=N$+STR$ (N-1
00*INT (N/100))
860 REM 508 NEXT J
865 REM 509 GOTO 320
870 REM 520 NEXT J
880 REM  530 GOTO 320
890 REM TRY WITH "59"; "959";
"9959", "99959", ... "999999959"
TO GET SCREENFULS OF RESULTS
900 REM TRY WITH "4294967255";
"4294967215"; ETC. TO FIND, SAY,
10 LARGEST PRIMES < 2**32
910 REM USE RANDOM INPUT TO SEE
HITHERTO UNSEEN PRIMES
920 REM EG PRESS RAND
930 REM THEN PRINT 5E7*(1+RND)
AND USE NEXT ODD NO. AS INPUT
940 REM TRY 5E8*(1+RND)
950 REM AND 2**31*(1+RND) TOO
960 REM SEE PRIMES NO-ONE HAS
EVER SEEN BEFORE
970 REM SAVE MODIFIED PROGRAM
980 REM GOOD LUCK
990 REM CAUTION THE ZX81 HOLDS

2**32-5 AND 2**31-1 INACCURATELY
1000 REM IT HOLDS 2**32-2-3 AND
(2**32-2)/2 ACCURATELY
```

HEX CODE FOR SPRF IN LINES 2 TO 6:-

```
D9  22  82  67  ED  53  80  67
D9  11  03  00  CD  E7  64  11
05  00  CD  E7  64  11  07  00
CD  E7  64  11  0B  00  CD  E7
64  13  13  CD  E7  64  13  13
13  13  CD  E7  64  13  13  CD
E7  64  13  13  13  13  CD  E7
64  21  06  00  19  EB  CD  E7
64  13  13  CD  E7  64  21  06
00  19  EB  CD  E7  64  13  13
13  13  CD  E7  64  13  13  CD
E7  64  13  13  13  13  CD  E7
64  21  06  00  19  EB  CD  E7
64  21  06  00  19  EB  CD  E7
64  13  13  CD  E7  64  21  06
00  19  EB  CD  E7  64  13  13
13  13  CD  E7  64  13  13  CD
E7  64  21  06  00  19  EB  CD
E7  64  13  13  13  13  CD  E7
64  21  06  00  19  EB  CD  E7
64  21  08  00  19  EB  CD  E7
64  13  13  13  13  CD  E7  64
13  13  CD  E7  64  13  13  13
13  CD  E7  64  13  13  CD  E7
64  13  13  13  13  CD  E7  64
21  08  00  19  EB  CD  E7  64
21  06  00  19  EB  CD  E7  64
13  13  13  13  CD  E7  64  21
06  00  19  EB  CD  E7  64  13
13  CD  E7  64  13  13  13  13
CD  E7  64  21  06  00  19  EB
CD  E7  64  13  13  CD  E7  64
21  06  00  19  EB  CD  E7  64
21  06  00  19  EB  CD  E7  64
13  13  13  13  CD  E7  64  13
13  CD  E7  64  13  13  13  13
CD  E7  64  21  06  00  19  EB
CD  E7  64  13  13  CD  E7  64
21  06  00  19  EB  CD  E7  64
13  13  13  13  CD  E7  64  13
13  CD  E7  64  13  13  13  13
CD  E7  64  13  13  CD  E7  64
21  0A  00  19  EB  CD  E7  64
13  13  CD  E7  64  21  0A  00
19  EB  CD  E7  64  C3  A0  63
D9  2A  90  67  ED  5B  92  67
D9  21  00  00  06  20  A7  D9
ED  6A  EB  ED  6A  EB  D9  ED
6A  38  08  ED  52  30  07  19
A7  18  04  A7  ED  52  37  10
EB  D9  ED  6A  EB  ED  6A  EB
D9  7C  B5  28  0B  A7  2A  94
67  ED  52  D0  3E  01  18  05
AF  ED  53  8E  67  32  96  67
D9  2A  82  67  ED  5B  80  67
D9  E1  C9  00  00  00  00  00
00  00  00  00  00  00  00  00
00  00  00  00  00  00  00  00
00  00  00  00  00  00  00  00
00  00  00  00  00  00  00  00
00  00  00  00  00  00  00  00
00  00  00  00  00  00  00  00
00  00  00  00
```





# Between the Stars

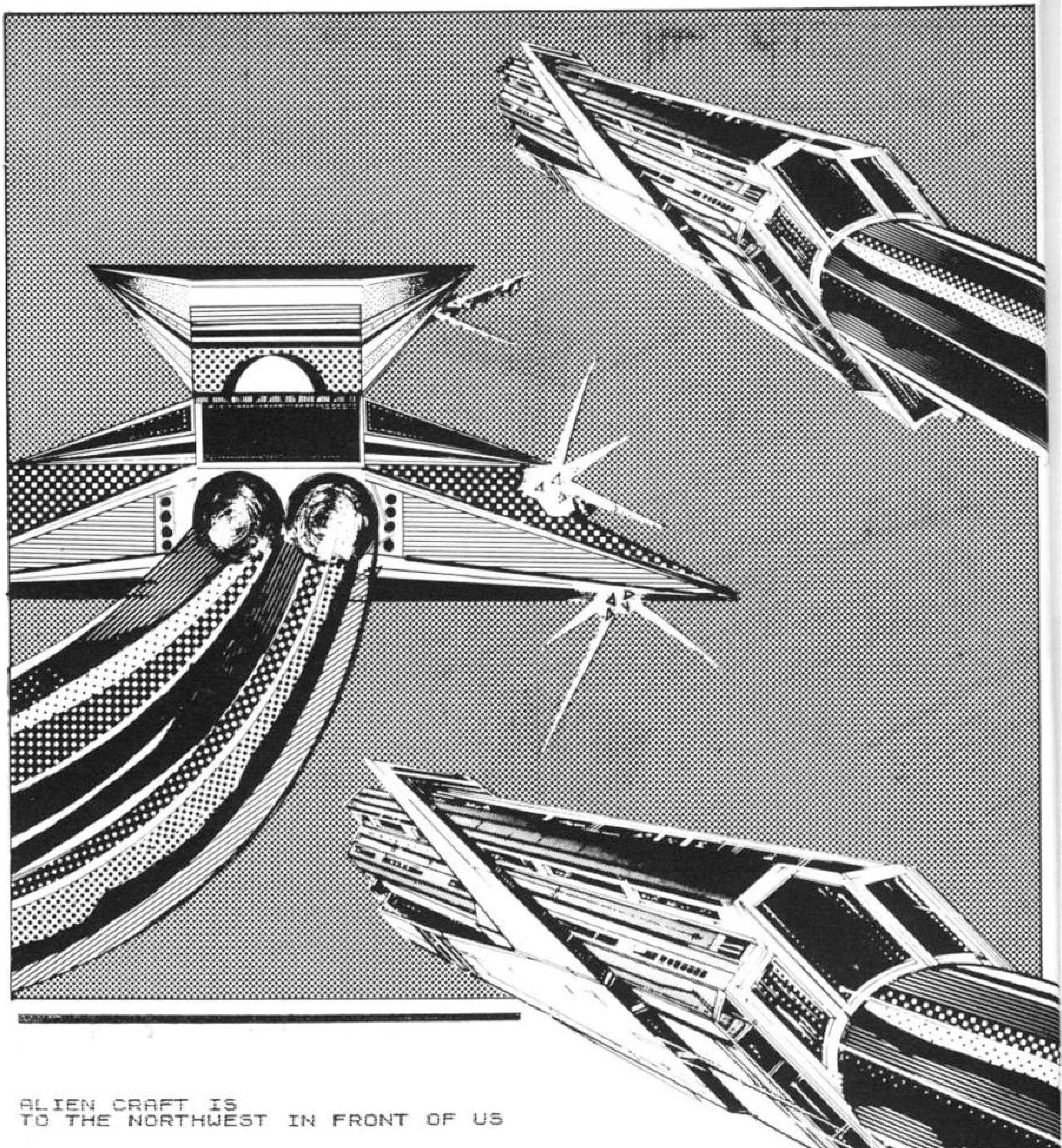
**Roger MacIntyre from Ravenscourt Park has decided the delights of West London are not enough for him. He prefers the space lanes, where he is responsible for the security of a cube of space, measuring 10 x 10 x 10. The Terran Federation, sparing no expense in the defence of earth, have provided him with a space ship equipped with a ZX81 as its on-board computer. Roger needs a break on earth, so now it is your task to guard the space lanes.**

The printout shows you what the screen looks like when you play this game. There are a lot of things demanding your attention. Your position within the cube is given by the three co-ordinates under the line "SHIP IS CRUISING AT CO-ORDINATES:". The first co-ordinate is your position north/south (with lower numbers to the south), the second is your position across the cube, ie east/west, and the third is your position within the cube (forward/back). You can see that the ability to visualise in three-dimensions is useful.

The alien craft is moving very slowly within the cube, but although you know, at all times, its direction from you, you do not know how far away it is. You have to hit it as many times as you can before the time counter decrements to zero, and without colliding with the alien craft. Running out of energy will also terminate the game. You will know when you are close enough to fire when the computer reports that the alien ship is firing at you. Every hit decrements your energy supply rather drastically.

Thje game is simple to play, despite the bewildering amount of input the program is giving you. You just touch the key which refers to the direction you want to move, N, S, E or W to move north, south, east or west, A to advance, R to retreat and L to fire your laser at the alien ship. If, for example, you knew the ship was to the north, you could just hold down the N key until you moved onto the same north/south plane as the ship, then test for proximity by firing.

You'll find that the program will teach you how to play the game. Just keep in mind that you have to get as close as possible to the alien ship to fire, and that your task is to get as many on your 'tally' as possible before the game ends.



```
ALIEN CRAFT IS
TO THE NORTHWEST IN FRONT OF US


ENERGY LEFT ■ 46 ERGS


■SHIP IS CRUISING AT THE
CO-ORDINATES:
    9  5  8
ENTER YOUR COMMAND
  N,S,E,W,(L)ASER,
(A)DVANCE,(R)ETREAT TIME: 34
                    TALLY: 0
```

```
10 REM BETWEEN THE STARS
20 REM BY ROGER MACINTYRE
30 GOSUB 1070
40 GOSUB 800
50 IF L<0 THEN GOTO 500
60 PRINT AT 17,0;"ENTER YOUR C
```

```
OMMAND"
 90 PRINT AT 18,2;"N,S,E,W,(L)A
SER,","(A)DVANCE,(R)ETREAT"
 100 LET L=L-0.25
 120 IF INKEY$="" THEN GOTO 120
 130 IF INKEY$="L" THEN GOSUB 32
0
 140 IF INKEY$="N" THEN LET X=X-
1
 150 IF INKEY$="S" THEN LET X=X+
1
 160 IF INKEY$="E" THEN LET Y=Y+
1
 170 IF INKEY$="W" THEN LET Y=Y-
1
 180 IF INKEY$="A" THEN LET Z=Z-
 190 IF INKEY$="R" THEN LET Z=Z+
1
 195 PRINT AT 5,0;S$
 200 GOSUB 620
 210 IF RND>0.5 THEN GOTO 40
 240 LET A=A+INT ((RND*3)-(RND*3

))
 250 IF A<1 THEN LET A=1
 255 IF A>10 THEN LET A=10
 260 LET B=B+INT ((RND*3)-(RND*3
))
 265 IF B>10 THEN LET B=10
 266 IF B<1 THEN LET B=1
 270 IF RND>0.5 THEN GOTO 40
 280 LET C=C+INT ((RND*3)-(RND*3

))
 290 IF C<1 THEN LET C=1
 300 IF C>10 THEN LET C=10
 310 GOTO 40
 320 REM ** FIRE LASER **
 330 LET L=L-0.75
 340 PRINT AT 1,0;
 350 IF ABS (A-X)>3 OR ABS (B-Y)
>3 OR ABS (C-Z)>3 THEN PRINT AT
1,0;"OUT OF RANGE..."
 360 FOR J=1 TO 50
 370 NEXT J
 375 PRINT AT 1,0;T$
 380 IF ABS (A-X)>3 OR ABS (B-Y)
>3 OR ABS (C-Z)>3 THEN RETURN
 390 PRINT AT 1,0;"ORDER TO FIRE
UNDERSTOOD"
 400 FOR J=1 TO 50
 410 NEXT J
 415 PRINT AT 1,0;T$
 420 IF RND<.65 THEN GOTO 470
 430 PRINT AT 1,6;"██████ MISSED

██████"
 440 FOR J=1 TO 50
 450 NEXT J
 455 PRINT AT 1,0;T$
 460 GOTO 490
 470 PRINT AT 1,0;"COMPUTER REPO
RTS ACCURATE HIT"
 480 LET T=T+1
 482 FOR J=1 TO 50
 483 NEXT J
 485 PRINT AT 1,0;T$
 490 RETURN
 500 PRINT
 510 PRINT TAB 3;"█ TERMINATION
█"
 520 PRINT

 530 IF TI<0 THEN PRINT "WE HAVE
BEEN IN SPACE TOO LONG"
 540 IF L>0 THEN PRINT "█ WE HAV
E BEEN DEFEATED █"
 550 PRINT
 555 PRINT AT 10,0;"ENERGY LEFT
█ ";L;" ERGS      "
 560 IF L<=0 THEN PRINT "███████
ENERGY BANKS EMPTY ██████"
 570 STOP
 580 PRINT
 590 PRINT "WE HAVE COLLIDED WIT
H THE";TAB 6;"ALIEN SHIP"
 610 STOP
 620 REM ** ALIENS SHOOT **
 630 IF ABS (A-X)>3 OR ABS (B-Y)
>3 OR ABS (C-Z)>3 THEN RETURN
 650 IF RND>0.75 THEN RETURN
 660 PRINT AT 1,0;"██████ ALIENS
FIRING AT US ████"
 670 FOR J=1 TO 50
 680 NEXT J
 690 PRINT AT 1,0;T$
 700 IF RND>0.7 THEN GOTO 770
 710 PRINT AT 1,0;"█ ALIEN FIRE
HAS HIT US █"
 720 LET L=L-7
 730 IF L<=0 THEN GOTO 500
 740 FOR J=1 TO 50
 750 NEXT J
 755 PRINT AT 1,0;T$
 760 RETURN
 770 PRINT AT 1,0;"██████ALIEN F
IRE MISSED███████"
 780 FOR J=1 TO 50
 790 NEXT J
 792 PRINT AT 1,0;T$
 795 RETURN
 800 REM ** PRINT OUT **
 850 PRINT AT 10,0;"ENERGY LEFT
█ ";L;" ERGS      "
 870 LET TI=TI-1
 880 IF TI=0 THEN GOTO 500
 890 PRINT AT 19,20;"TIME: ";TI
 900 IF L<3 THEN PRINT AT 12,4;"
██ ENERGY LOW ██"
 920 PRINT AT 20,19;"TALLY: ";T
 930 PRINT AT 14,0;"█SHIP IS CR
UISING AT THE"
 935 PRINT "CO-ORDINATES:"
 936 PRINT TAB 4;X;"   ";Y;"   ";Z
;""
 940 IF A=X AND B=Y AND C=Z THEN
GOTO 580
 960 PRINT AT 5,0;"▀▄";AT 5,0;"▀▄
▄";AT 5,0;""
 967 PRINT AT 5,0;"ALIEN CRAFT I
 970 IF A<>X OR B<>Y THEN PRINT
"TO THE ";
 980 IF A<X THEN PRINT "NORTH";
 990 IF A>X THEN PRINT "SOUTH";
1000 IF B>Y THEN PRINT "EAST";
1010 IF B<Y THEN PRINT "WEST";
1020 IF C=Z THEN PRINT " OF US"
1030 IF C>Z THEN PRINT " BEHIND
US"
1040 IF C<Z THEN PRINT " IN FRON
T OF US"
1060 RETURN
1070 REM ** INITIALISE **
1090 LET L=25+INT (RND*30)
1100 LET T=0
1110 LET TI=35
1140 LET A=INT (RND*10)+1
1150 LET B=INT (RND*10)+1
1160 LET C=INT (RND*10)+1
1170 LET X=INT (RND*10)+1
1180 LET Y=INT (RND*10)+1
1190 LET Z=INT (RND*10)+1
1195 LET S$="

        "
1197 LET T$="
        "
1200 FOR J=0 TO 63
1205 PLOT J,0
1210 PLOT J,43
1220 NEXT J
1230 FOR J=0 TO 4
1240 PLOT 0,J
1250 PLOT 63,J
1260 NEXT J
1270 RETURN
```

# String along with your friends

## Graham Charlton from Romford has contributed some fine utility programs for the ZX81.

### Telephone Directory

When you run this program, you'll be given three options — update, search or save. Pressing 1 (update), enables you to add to your directory. It asks for the name of the person you wish to enter, and then the number. This is converted (see line 1080) to a 32 character length string. It is then placed into your growing directory in alphabetical order (1190-1160). The program then requests another name. Simply pressing NEWLINE returns you to the three options.

Entering "2" (search) allows you to search for the number required. Enter the name of the person whose number you want to find, and it will search for this name, and print it out. You can have two or more entries for one person, ie. home and work numbers, the program will print out all of them. If you enter "A" then all the names and numbers of the people whose name begins with A in your directory will be printed out. If you enter BA you'll get all the names starting with BA, and so on. Entering a null string will print out the whole directory in alphabetical order. Pressing "3" (save) saves the enlarged database.

```
  10 GOTO 3000
1000 SCROLL
1010 PRINT "NAME TO BE ENTERED?"
1020 INPUT M$
1030 IF M$="" THEN RETURN
1040 SCROLL
1050 PRINT M$;"S NUMBER?"
1060 INPUT N$
1070 LET W=W+32
1080 LET A$=A$+M$+"..............
......................"( TO 32-LEN
(M$+N$))+N$
1090 LET X=W
1100 FOR Y=X TO 1 STEP -32
1110 IF A$(X TO X+31)>A$(Y TO Y+
31) THEN GOTO 1000
1120 LET B$=A$(Y TO Y+31)
1130 LET A$(Y TO Y+31)=A$(X TO X
+31)
1140 LET A$(X TO X+31)=B$
1150 LET X=Y
1160 NEXT Y
1170 GOTO 1000
2000 SCROLL
2010 PRINT "NAME TO BE FOUND?"
2020 INPUT N$
2030 FOR Z=1 TO W STEP 32
2040 IF A$(Z TO Z+LEN N$-1)<>N$
THEN GOTO 2080
2050 SCROLL
2060 PRINT A$(Z TO Z+31)
2070 IF INKEY$="Q" THEN PAUSE 4E
4
2080 NEXT Z
2090 SCROLL
2100 PRINT "SEARCH COMPLETED"
2110 RETURN
```

```
3000 SCROLL
3010 PRINT TAB 7;"TELEPHONE DIRE
CTORY"
3020 SCROLL
3030 PRINT TAB 10;"BY G.CHARLTON
"
3040 LET A$=""
3050 LET W=-31
3060 SCROLL
3070 SCROLL
3080 PRINT "1-UPDATE        2-SEARC
H     3-SAVE"
3090 SCROLL
3100 LET Z$=INKEY$
3110 IF Z$="" THEN GOTO 3100
3120 IF Z$="1" THEN GOSUB 1000
3130 IF Z$="2" THEN GOSUB 2000
3140 IF Z$="3" THEN SAVE "TELEPH
ONE DIRECTORY"
3150 GOTO 3060
```

### String Sort

The title should give away what this program does. You are asked how many words you wish to enter, and the maximum length of the words. This sets up a two dimensional strsets up a two dimensional string array.

You then enter the words, the ZX81 switches into FAST and sorts the words into alphabetical order, switches into SLOW, and prints out the list. To print the list onto paper, delete line 250 and change line 260 to LPRINT A$(A).



```
 10 PRINT "NUMBER OF WORDS TO B
E SORTED?"
 20 INPUT W
 30 PRINT "MAXIMUM LENGTH OF WO
RDS?"
 40 INPUT L
 50 DIM A$(W,L)
 60 FOR A=1 TO W
 70 INPUT A$(A)
 80 NEXT A
 85 FAST
 90 LET A=0
100 LET B=A

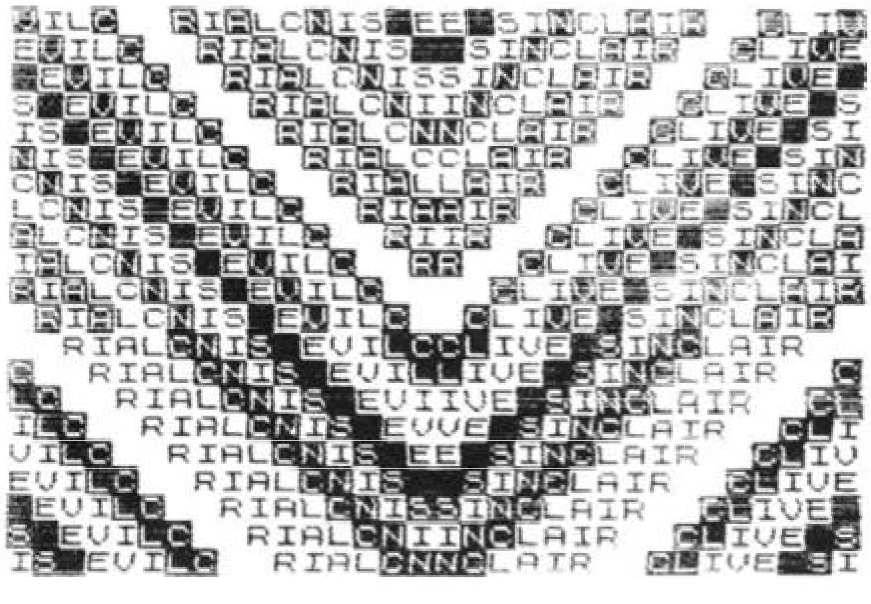
110 LET C=A
120 LET C=C+1
130 IF C=W THEN GOTO 235
140 LET A=C
150 LET B=A+1
160 IF A$(B)<A$(A) THEN LET A=B
170 LET B=B+1
180 IF B<=W THEN GOTO 160
190 IF C=A THEN GOTO 120
200 LET B$=A$(A)
210 LET A$(A)=A$(C)
220 LET A$(C)=B$
230 GOTO 120
235 SLOW
240 FOR A=1 TO W
250 SCROLL
260 PRINT A$(A)
270 NEXT A
```

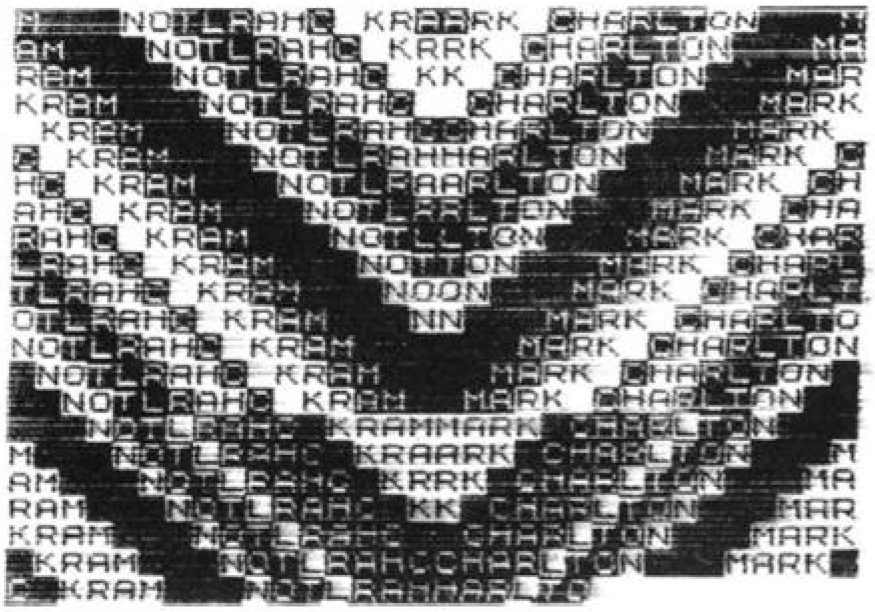
## Wallpaper

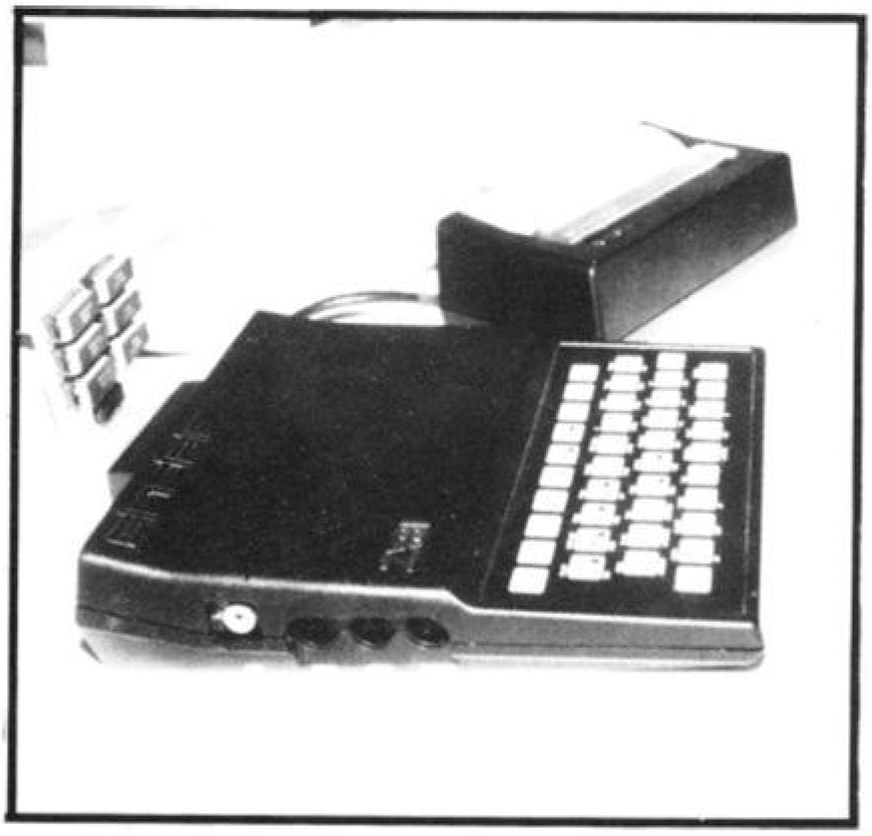
From Mark Charlton comes a program which expects you to enter a name, some words, or a design, and then from the string you enter attempts to create 'wallpaper'. Sample runs follow the program, using the words 'MARK CHARLTON', 'CLIVE SINCLAIR' and 'ZX COMPUTING'. Mark suggests you could try it just by pressing NEWLINE, without entering anything, which still produces a fine design, or just use a few graphics symbols and spaces.

```
 10 REM NAME WALLPAPER
 20 REM (C) MARK CHARLTON 1982
 25 SCROLL
 30 PRINT "ENTER YOUR NAME"
 35 SCROLL
 40 INPUT A$
 45 LET A$=A$+" "
 46 IF LEN A$<16 THEN GOTO 45
 47 LET A$=A$( TO 16)
 50 FOR G=1 TO 16
 60 IF RND>=.5 AND CODE A$(G)<1
28 THEN LET A$(G)=CHR$ (CODE A$(
G)+128)

 70 IF RND>=.5 AND CODE A$(G)>1
27 THEN LET A$(G)=CHR$ (CODE A$(
G)-128)
 80 NEXT G
120 FOR H=1 TO 16
130 FOR A=-16 TO 16
145 IF A=0 THEN GOTO 160
150 PRINT A$(ABS A);
160 NEXT A
170 SCROLL
180 LET A$=A$(2 TO )+A$(1)
190 NEXT H
200 GOTO 50
```







# Dot-dot-dot, dash-dash-dash

## Master Morse code with the help of this 16K ZX81 program from John Knight of Cheshire.

One of the conditions for getting an amateur radio licence (Class A UK) is a degree of proficiency in Morse code. This program may help you attain the required level of skill.

When you run the program a menu will appear giving you the option of entering an English message, and having it reprinted in Morse, of having the program generate a Morse symbol at random and give you three tries at entering its English equivalent.

Notice the use of the initialisation subroutine starting at line 9000, which goes into FAST, then strips A$ down to elements of C$. To simplify later processing, C$(38) is the equivalent of CHR$(38), ie. the letter "A". The program tells you (line 2190) which letter a particular symbol represents if you don't guess it within the three guesses allowed.

```
  10 REM MORSE TRAINER
  20 REM (C) J KNIGHT,1982
  30 GOSUB 9000
  40 FOR G=1 TO 10
  41 SCROLL
  42 NEXT G
  44 PRINT "MAKE A SELECTION:"
  45 SCROLL
  46 SCROLL
  47 SCROLL
  48 PRINT "1 - ENGLISH TO MORSE
"
  50 SCROLL
  55 SCROLL
  60 PRINT "2 - MORSE TO ENGLISH
"
  70 SCROLL
  75 SCROLL
  80 PRINT "3 - TO END"
  90 INPUT T
 100 GOSUB T*1000
 110 GOTO 40
1000 REM ENGLISH TO MORSE
1002 SCROLL
1003 SCROLL
1005 SCROLL
1010 PRINT "ENGLISH TO MORSE"
1015 SCROLL
1017 SCROLL
1020 PRINT "ENTER YOUR MESSAGE,
THEN"
1025 SCROLL
1030 PRINT TAB 3;"PRESS NEWLINE"
1040 INPUT W$
1045 SCROLL
1050 FOR G=1 TO LEN W$
1055 IF W$(1)<>" " THEN GOTO 108
0
1060 SCROLL
1065 SCROLL
1070 GOTO 1090
1080 PRINT C$(CODE W$(1));
1090 LET W$=W$(2 TO )
1100 NEXT G
1120 IF INKEY$="" THEN GOTO 1120
1130 RETURN
2000 REM MORSE TO ENGLISH
2002 SCROLL
2005 SCROLL
2007 SCROLL
2010 PRINT "I WILL GIVE YOU A LE
TTER IN"
2015 SCROLL
2020 PRINT "MORSE, AND YOU HAVE
THREE"
2025 SCROLL
2030 PRINT "GUESSES TO WORK OUT
WHAT IT IS."
2035 SCROLL
2040 SCROLL
2050 PRINT "PRESS NEWLINE WHEN Y
OU ARE"
2055 SCROLL
2060 PRINT TAB 3;"READY TO START
"
2070 IF INKEY$="" THEN GOTO 2070
2075 LET S=0
2080 FOR Z=1 TO 10
2090 LET J=38+INT (RND*26)
2100 SCROLL
2110 PRINT "WHAT LETTER DOES ";C
$(J)
2120 SCROLL
2130 PRINT TAB 12;"REPRESENT?"
2140 FOR H=1 TO 3
2150 INPUT K$
2160 IF CODE (K$)=J THEN GOTO 22
20
2170 SCROLL
2175 SCROLL
2180 IF H<3 THEN PRINT "NO, TRY
AGAIN"
2190 IF H=3 THEN PRINT C$(J);" R
EPRESENTS ";CHR$ (J)
2200 NEXT H
2210 GOTO 2250
2220 SCROLL
2230 PRINT "YES, YOU ARE RIGHT"
2240 LET S=S+1
2250 SCROLL
2255 SCROLL
2260 PRINT "YOUR SCORE IS ";S;"
OUT OF ";Z
2265 SCROLL
2270 SCROLL
2280 PRINT "▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒"
2290 SCROLL
2300 SCROLL
2310 NEXT Z
2320 RETURN
3000 STOP
9000 FAST
9005 LET A$=".-*-...*-.-.*-..*.*
..-.*--.*....*..*.---*-.-*.-..*--*-
-*-.*---*.--.*--.-*.-.*...*-*..-
*...-*.--*-..-*-.--*--..**"
9010 DIM C$(64,5)
9020 FOR B=38 TO 64
9030 LET B$=""
9040 FOR C=1 TO 5
9045 IF A$(1)="*" THEN GOTO 9080
9050 LET B$=B$+A$(1)
9060 LET A$=A$(2 TO )
9070 NEXT C
9080 LET C$(B)=B$
9085 LET A$=A$(2 TO )
9090 NEXT B
9100 SLOW
9500 RETURN
```

## WHAT CAN I DO WITH 1K?

If your answer is "not much", then you must read the new book from V&H, 'What Can I Do With 1K? (40 programs and routines for the 1K Sinclair ZX81)'.
"...a splendid book and one which will repay your investment time and time again." (ZX Computing).

Book £4.95.
Also available on cassette £4.95.

**NEW**

## WHAT CAN I DO WITH 16K?

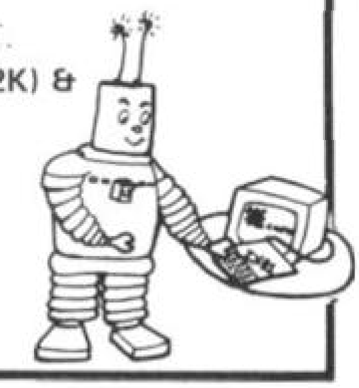
The companion volume to the above, containing complete program listings for 16K.

Book £4.95.
Cassettes available individually — ask for list.

## ZX81 PAYROLL

Probably the best ZX program ever written for the serious business user.
Cassette (only) £12.65 inc VAT
Manual (only) £2.00
Cassette & Manual £14.50 inc VAT.

**ALSO AVAILABLE FOR PET (32K) &**



**V&H Computer Services**
**182c Kingston Rd.**
**Staines**
**Middx.**
**Tel: Staines 58041**

**ZX81 QUALITY SOFTWARE**
**FOR THE SERIOUS USER**

# VIDEO-INDEX

**BUILD YOUR OWN INDEX OF UP TO 1000 ENTRIES ON YOUR 16K ZX81**

- Designed specifically for the ZX81.
- Over 1000 references possible in 16K RAM.
- Up to 57 characters of text generated for each reference.
- Ingenious machine coded encryption technique.
- Fast and powerful machine coded search procedure.
- Minimum keystroke data entry procedure.
- Menu driven, easy to operate and crash proofed.
- Documented to the usual high Video Software standard.
- Includes demonstration index of ZX81 magazine references.

This is probably the best serious program yet written for the ZX81 and is available from good software retailers or by mail order direct from ourselves.

**Price £9.95 incl. VAT**
**Plus £0.55 P&P**

**Send large SAE for our full catalogue of software products.**

**Video Software Ltd**
**Stone Lane Kinver**
**Stourbridge**
**West Midlands DY7 6EQ**

# THE EXPLORER'S GUIDE To The ZX81

The Book for the ZX81 Enthusiast.

By Mike Lord, 120 pages.

Programs for 1K RAM, and programs for 16K RAM. Games, Business and Engineering Applications. RAM & I/O Circuits. Useful ROM Routines. Hints and Tips.

£4.95

## What Can I Do with 1K?

By Roger Valentine. A fresh and original book containing 40 programs and routines for the unexpanded ZX81.

£4.95

## The ZX80 Magic Book

"With 8K ROM/ZX81 Supplement"

£4.75

## Mastering Machine Code on your ZX81

By Toni Baker. 180 pages of immense value to beginner and expert alike.

£7.50

We welcome Access

VISA

ALL PRICES INCLUDE U.K. P & P AND 15% VAT WHERE APPLICABLE. OVERSEAS CUSTOMERS ADD £1.50 CARRIAGE PER ORDER.

TIMEDATA LTD Dept G 57 Swallowdale, Basildon, Essex SS16 5JG Tel: (0268) 411125 (MON-FRI)

**TIMEDATA**

# SECOND FOUNDATION

## ZX81 SOFTWARE

ALL OUR SOFTWARE
IS ON CASSETTE AND DELIVERED BY RETURN

'COMPUTATUNE' TURNS THE 1K ZX81 INTO A ONE OCTAVE ORGAN, STORING 100 NOTES. PLAYS BACK TUNE AUTOMATICALLY - £2.75

'STARTER PACK' TWELVE 1K GAMES ON ONE CASSETTE, SEVEN MOVING GRAPHIC GAMES INCLUDING - Road-Race, Subsearch, Tank-Shoot AND Pop. EXCELLENT VALUE - £3.90

'ARCADE PACK' FOUR 16K GAMES ON ONE CASSETTE - Surround, Tank-Shoot, Spider Chase AND Tenpin Bowling - £3.90

'VARIED GAME PACK' FOUR 16K GAMES, Jaywalker, Car-Crash, Smargana AND Fox and Hounds. "CAR-CRASH" IS MACHINE CODE AND WORTH £3.90 ALONE! ALL FOUR GAMES - £3.90

'TOLINKA' - 16K. USE YOUR ZX81 AS A CHESS GAME RECORDER. FEATURES TOO NUMEROUS TO MENTION - £3.90

'G.C.E. EXAMS' 180 MIN. EXAM IN 'O' LEVEL PHYSICS OR MATHS - GIVES GRADE AND DETAILED REVISION PROGRAMME - £3.90 EACH.

Prices are inclusive (Overseas add £1.00 P&P)
Send s.a.e. for details, or cheque with order to:
SECOND FOUNDATION, 22 Bramber, Belgrave, Tamworth, Staffs. B77 2LL

# First steps in ZX BASIC

**Your first hours with a ZX Computer can be bewildering as you try to make sense of the manual, and sort out just what you can do with your new possession. Mark Charlton, author of** The Gateway Guide to the ZX80 and the ZX81, **discusses some of the fundamental parts of the BASIC programming language. Although the program printouts are from a ZX81, all the material here applies to the ZX Spectrum, and most of it to the ZX80.**



shown in the manual, then typing in SAVE followed by the name of the program within quote marks. In this case, I suggest you use the name ROCKET, so you would type in SAVE "ROCKET". Turn your cassette recorder on to record, after connecting it up as shown in the manual, and then press the NEWLINE/RETURN key.

I suggest you make a habit of saving each program three times in a row, on a C-12 or C-15 (ie. computer) cassette, and that you only put one program on each side of a tape. Label the tape clearly with the load name (ie. with ROCKET in this case).

Although it may seem wasteful to use up the whole side of a cassette with just one program recorded three times, the frustration you will save yourself by not having to search through tape after tape for a program you want will more than compensate for using more cassettes than is strictly necessary. The program is recorded three times just in case the tape gets damaged at some point, or you accidentally erase part of the program, or — as sometimes happens — one recording of the program refuses to load properly.

You should clean the recorder's heads frequently using liquid (not a tape cleaner ribbon in a cassette) to ensure the clearest possible signal is put onto the tape.

## Scientific notation

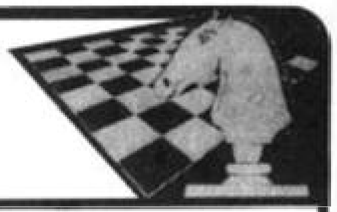
Finally, in this article, we'll have a look at scientific notation. A computer uses what is known as scientific notation to display large numbers as a single digit and up to eight decimal places, followed by the letter E (for exponention) and the power of 10 to which the number is to be multiplied. Enter and run PROGRAM 12 (SCIENTIFIC NOTATION) which shows a variable (A) assigned to a number (1234) in line 20, then repeatedly printed out, then multiplied by 10. You can see (Fig. 2) part of the print out underneath the program listing.

Note that after the number has nine trailing zeroes (1234000000000) it is printed as a number, a decimal point, more numbers after the decimal point, the letter E and a power of 10. Try and predict how long this program will run until it exceeds the maximum number possible on a ZX computer, then run it until it crashes to see if you were right.



**moving ahead with ZX software**

# ZXCHESS



## ZX CHESS & ADVENTURES

PROGRAMS FOR THE ZX81/80 INCLUDING –

**ZX-FORTH** — Full Implementation of forth on the ZX. forth runs 10 to 26 times faster than BASIC "Simplicity of BASIC with the speed of machine code"

**BUG–BYTE RAM PACK**
16K – £29.50
32K – £42.95
64K – £65.95

No wobble problems
1 Year Guarantee on each Ram Pack. Simply the best you can buy. Immediate delivery.

**1K ZXCHESS! £2.95** — We didn't think it was possible but this actually plays against you. Two opening moves.

**ADVENTURES**

**ADVENTURE 'A' £6.00** — Exciting machine code games with instant response, choose from the range below. You find yourself stranded on an alien planet. Can you reach your ship and escape?

**ADVENTURE 'B' £7.00** — In a jungle clearing you come across an Inca temple. You must break in, collect treasure and escape alive. Beware. Includes a cassette save routine.

**ADVENTURE 'C' £8.00** — You are unfortunate enough to be drawn to an alien cruiser. Can you reach the control room and free yourself or will they get you first? Includes a cassette save routine.

**GALAXIANS £3.95**
All the features of the arcade game in a fast machine code program. Swooping attackers, explosions and personalised scoring.

**ZXBUG £7.00**
A 30 in 1 machine code tool and disassembler, allows access to registers and search through and modify memory; with cassette routines.

## Tool Kit (Programmers) £5.95

at 9 New Functions to the ZX81 Commands making programming easier.

## TWO GREAT ZX81 16K CHESS GAMES

### ZXCHESS (ENHANCED)

- Written totally in machine code.
- Full graphic display of Chess board.
- Six levels of play: Two play within competition time limits.
- Option to play Black or White.
- Plays all legal moves including castling and en-passant.
- Cassette routines for saving unfinished game and returning to later.
- Displays moves of game on screen or printer for analysis.
- Print a copy of the Chess board onto the printer.
- Board can be set up in any position, you can even swap sides midgame.
- Clear whole board with one command: for end game analysis.

**£6.50**

### ZXCHESS II

We believe the strongest ZX81 Chess game as no other has beaten it!!

- All the features of ZXCHESS plus:
- 32 opening moves.
- Seven levels of difficulty: FOUR play within competition time limits.
- A move is suggested by the ZX81 if wanted.
- Optional Full Graphic version using the QS CHRS Board.

**£9.95**

2 copies supplied on cassette with full instructions.
Cheques and postal orders payable to:

ARTIC COMPUTING LTD Dept AS1
396 JAMES RECKITT AVENUE,
HULL HU8 0JA

---

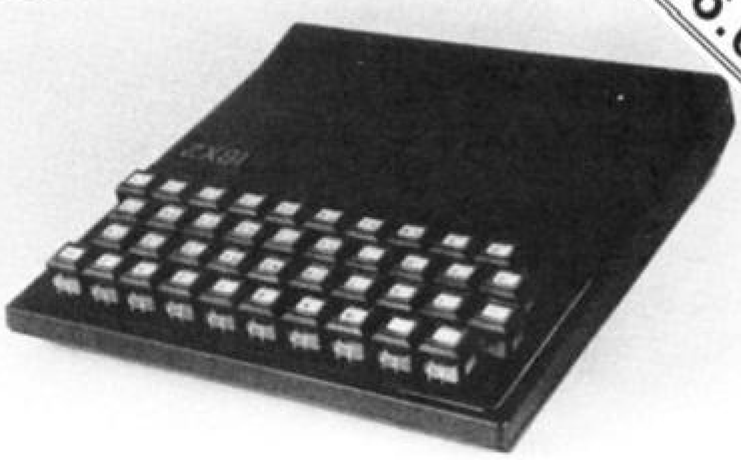
**NEW**

**£26.00**

## KEMPSTON (MICRO) ELECTRONICS

introducing the

# ZX81 KLIK-KEYBOARD



This is a full, forty key, moving keyboard that fits in the recess left after peeling off the existing 'touch sensitive' keypad.

Consider the following advantages

- POSITIVE feedback from keys
- Fits ONTO the ZX81
- NO trailing wires
- NO special case needed
- Elegent design with two colour legends.

The fully built keyboard requires ABSOLUTELY NO SOLDERING, as the keyboard is supplied with flexible connectors which simply plug into the existing sockets.

Alternatively, the keyboard is available as an easy to build kit at a considerable saving in cost.

Other ZX81 products available include a user port with 16 I/O lines at £16.50 built, keyboard bleeper which fits inside the case at £8.95 built and a repeat key kit at £3.95.

Proprietor A. Pandaal B.Sc. P.G. cert. Ed.

Date ____________

Your Name and Address ____________

| Quantity | Description | Unit Price | Amount |
| --- | --- | --- | --- |
| | ZX81 Keyboard kit | 22 50 | |
| | ZX81 Keyboard assembled | 26 00 | |
| | | Sub total | |
| | | Carriage | – 70 |
| | | Total due | |

Cheques/P.O. made payable to
Kempston Electronics
60 Adamson Court Hillgrounds Rd
Kempston, Bedford, MK42 8Q2
Allow 14 days delivery.

# The PLOT thickens

**Many dialects of BASIC include an ELSE option, used in the statement IF...THEN... ELSE. There is no such function in ZX BASIC, but the computer's logic can be used to emulate this. Wilton J. Faberge shows you how.**

The IF...THEN...ELSE is a very useful variation on IF. The computer can be programmed to do something if the condition being tested for is found to be true, and something else, other than just go to the next line, if the condition is found to be false.

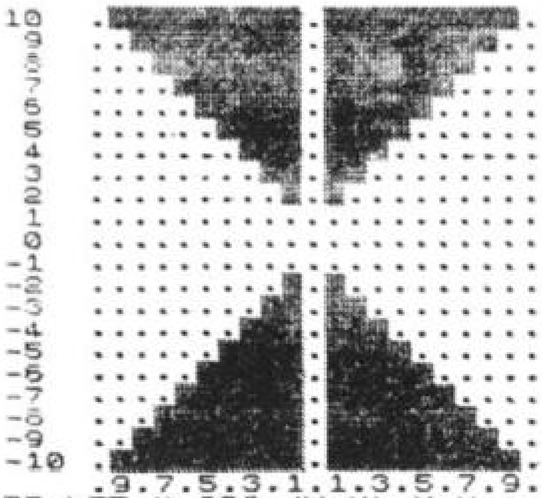
You can use the following substitution for IF...THEN... ELSE to produce some very interesting graphs. You simply enter the function you would like graphed in line 55. This is not the most efficient method of programming on the ZX computers, but it is useful as a means of demonstrating the IF...THEN...ELSE substitution.

As the program runs, it evaluates K each time it comes to line 55. Line 70 looks at the value of K and prints a zero if K is greater than or equal to point five, and a full stop if K is less than point five. This is the same as a line reading IF K is greater than or equal to point five print "0" ELSE print ".".

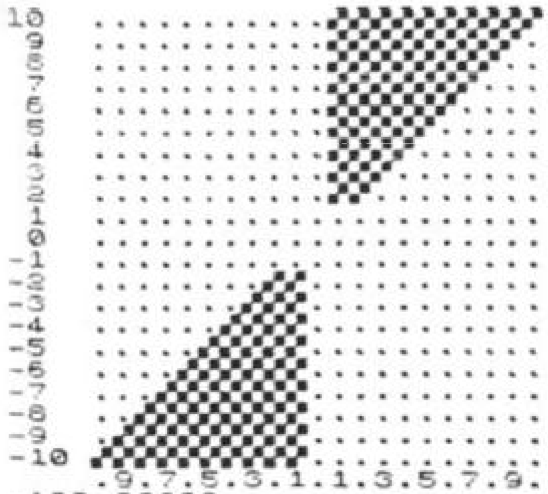
Each of the other graphs uses different values for K, as generated by line 55. The condition tested for in line 70 also varies. Run the samples given, using your own choice of graphics symbol in line 70, and then create a few of your own. It is likely that you'll have to change the scaling for certain functions.

```
 10 REM GRAPH-PLOTTER
 20 REM (C) W. J. FABERGE
 30 FOR Y=10 TO -10
    STEP -1
 35 IF Y<>10 AND Y<>
-10 AND Y>-1 THEN
    PRINT " ";
 40 PRINT Y;TAB 4;
 50 FOR X=-10 TO 10
 55 LET K=Y-X*X/2+7
 70 PRINT ("0" AND K>=.5)
+("." AND K<.5);
110 NEXT X
120 PRINT
130 NEXT Y
140 PRINT TAB 4;".
    9.7.5.3.1
   .5.7.9."
```

```
 10 .....00000000000.....
  9 .....00000000000.....
  8 .....00000000000.....
  7 .....00000000000.....
  6 .....00000000000.....
  5 ......000000000......
  4 ......000000000......
  3 ......000000000......
  2 ......000000000......
  1 .......0000000.......
  0 .......0000000.......
 -1 .......0000000.......
 -2 .......0000000.......
 -3 ........00000........
 -4 ........00000........
 -5 .........000.........
 -6 .........000.........
 -7 .....................
 -8 .....................
 -9 .....................
-10 .....................
    .9.7.5.3.1.1.3.5.7.9.
```

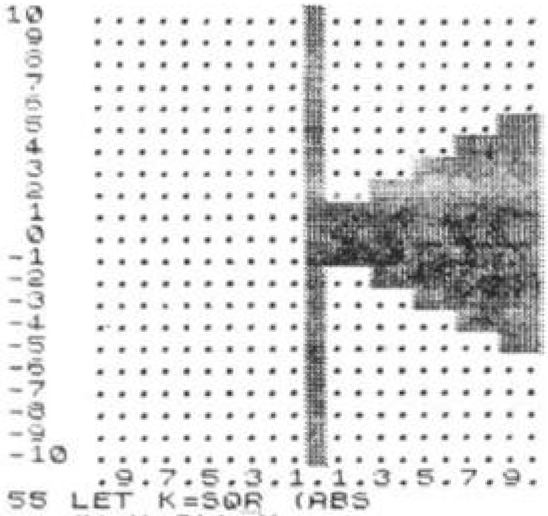


```
55 LET K=ABS (Y*X)-X*X
REVERSE THE GREATER
AND LESS
THANS FROM PREVIOUS EXAMPLE
```

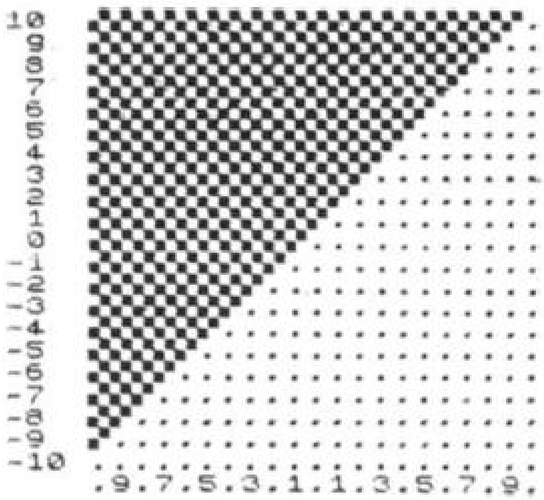


```
-190.90909
55 LET K=Y*X-X*X/1.1
70 PRINT ("▞"AND K>=.25)
+("." AND K<.25);
```

```
 10 .....00000000000.....
  9 .....00000000000.....
  8 ......000000000......
  7 ......000000000......
  6 ......000000000......
  5 .......0000000.......
  4 .......0000000.......
  3 ........00000........
  2 ........00000........
  1 .........000.........
  0 .....................
 -1 .........000.........
 -2 ........00000........
 -3 ........00000........
 -4 .......0000000.......
 -5 .......0000000.......
 -6 ......000000000......
 -7 ......000000000......
 -8 ......000000000......
 -9 .....00000000000.....
-10 .....00000000000.....
    .9.7.5.3.1.1.3.5.7.9.
3*ABS (Y)-X*X
```

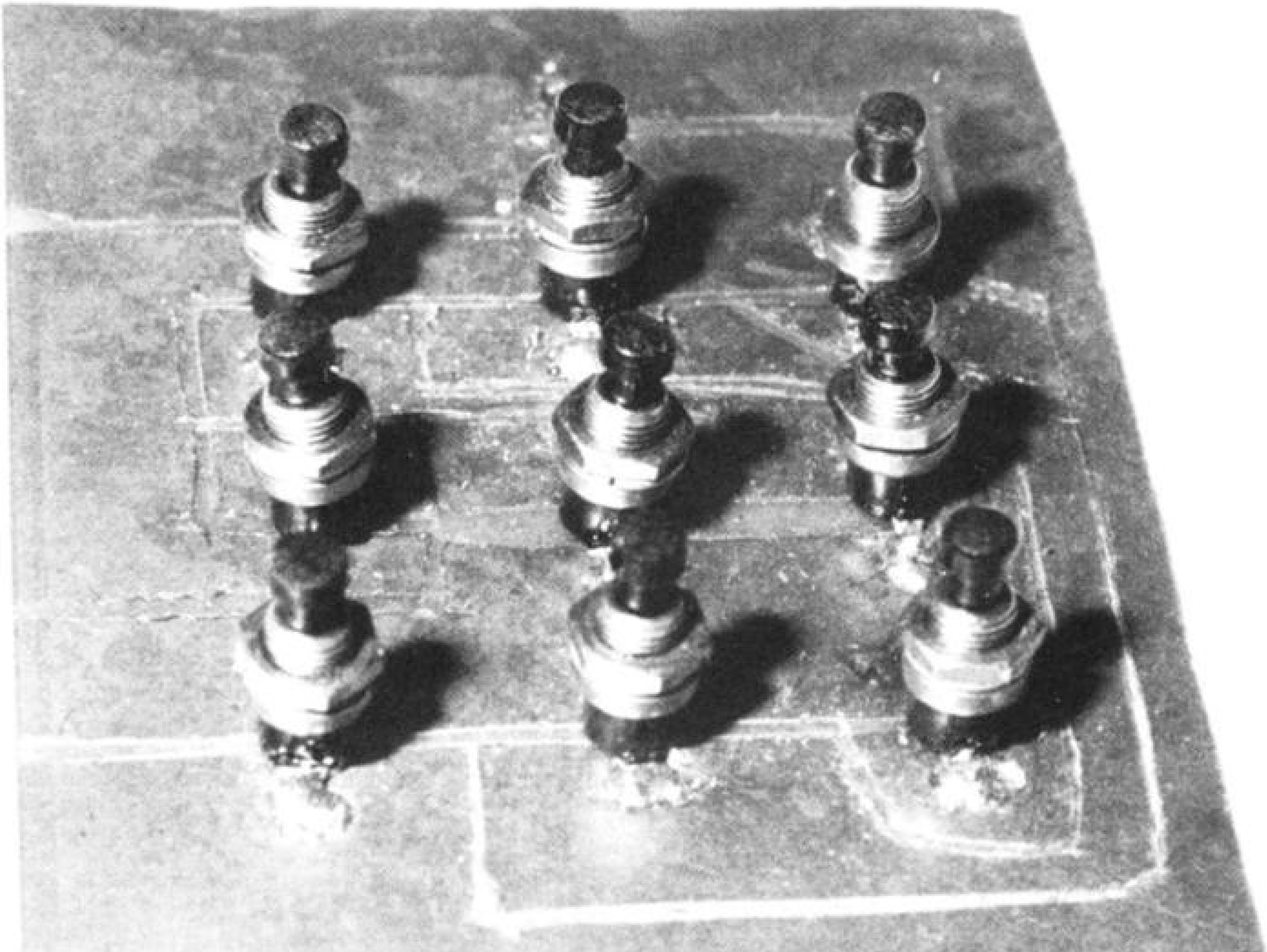


```
55 LET K=SQR (ABS
   (Y*X*2))-X
70 PRINT ("▒" AND K<=.5)
+("." AND K>.5);
```



# Adding a numeric keypad

**If your ZX81 is employed for business or mathematical use, you'll find this project — a numeric keypad — a worthwhile one to build. Taken from the book "20 Simple Electronic Projects for the ZX81", by Stephen Adams, this article discusses the role of an INPUT/OUTPUT port, and then explains how to use this information to help you build a numeric keypad for your ZX81.**



| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |

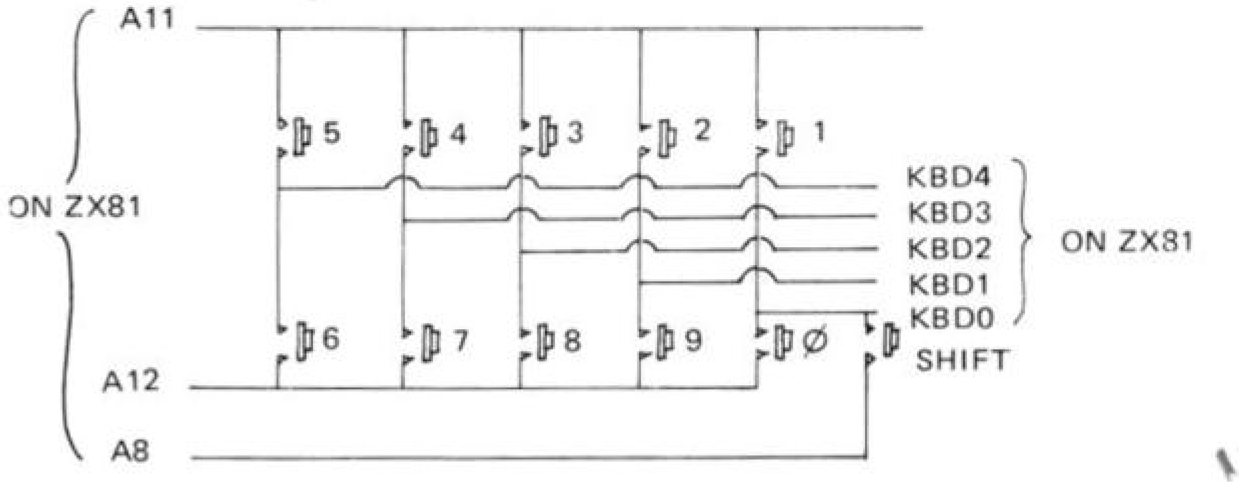
In order to make the ZX81 more useful, and allowing it to control things, we must first be able to send signals to and from the ZX81. A device to do this is called an INTERFACE.

A common interface is an INPUT/OUTPUT port, this consists of one or several chips which will store any data sent to it and keep it available for an external device. It will also allow you to "see" through it to an external device. The maximum amount of data that it can store is eight Binary (TWO STATE) BITS, which consist of eight wires which have either + 5 volts (binary 1) or 0 volts on them. The INPUT and OUTPUT ports are usually separate, so the data emitted by the output port is not affected by "reading" the input port. To tell if it is a READing operation or a WRITEing (OUTPUT) operation the ZX81 puts out two signals NOT WRITE (WR) and NOT READ (RD). The fact that either of these signals is at 0 volts (Binary 0), enables the operation to be done.

The device also requires a place whe. ı you know that you can collect and send your data. It is called an ADDRESS. The address applies only to this port and no other piece of equipment connected to the computer. The ADDRESS wires A0-A15 contain this number when the ZX81 wants to talk to your port.

There are several companies which produce INPUT/OUTPUT ports for the ZX81, but their ports fall into one of two categories.

One of these requires a special machine code routine to be written in order to get the data to and from the port. This is because they are treated differently to a normal memory location. They are in a separate memory map to the RAM (Random Access Memory), controlled by a signal called NOT INPUT/OUTPUT REQUEST (IOREQ). When this line is at 0 volts ALL memory is switched off the memory map and




replaced by locations numbered 0-255. Thus on INPUT/OUTPUT signals only ADDRESS lines A0-A7 need to be used. BUT because this is not available through BASIC, a special machine code routine needs to be written.

The other type of port is a MEMORY MAPPED port, which is treated like a piece of RAM. It may be PEEKd (transferred from the port into the program) or POKEd (transferred to the port from the program).

The ZX81 is not supplied with a users port, so one must be externally attached.

The PEEK and POKE are BASIC commands and can be included into a program in the following form:

```
PEEK 16396
POKE 16396,255
```

PEEK returns the number between 0 and 255 (the maximum number of combinations available from 8 bits). POKE puts a number between 0 and 255, which is after the comma, into the location in memory which is before the comma. No matter what method you use, you can only put in a number between 0 and 255. This is because we only have 8 bits (1's or 0's) at each location. These are numbered Bit 0 (B0) to Bit 7 (B7), and shown in Fig. 1).

Each bit represents a number in the multiplication table. The bit number gives the number of times 2 must be multiplied by itself, if it contains a BINARY 1(1), ie. if Bit 3 is Binary 1 then it represents 2 x 2 x 2 or 8. If it is Binary 0(0), then it represents exactly that 0. One thing to watch out for is Bit 0, when it is Binary 1, represents an odd number eg. 1. An example is that, if Bit 7 and Bit 0 are Binary 1 and the rest are Binary 0, it equals 128 + 1 (129). Try this for yourself with different numbers from bits to numbers and back again.

If you have trouble with converting numbers into bits then try this. Subtract the highest number below yours scoring a Binary 1 in this bit. Then do it again until you reach 0.

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| Number 28 | Bit 5 = 1 | 4 | Bit 2 = 1 |
| −16 | | −4 | |
| 12 | | Ø | |
| −8 | Bit 3 = 1 | Number 28 = 00101100 | |

All the rest of the eight bits must therefore be 0's.

The ZX81 keyboard is a matrix of switches which each connect ONE address line and ONE data line input. As there are five data inputs (KBD0-KBD4) and eight address lines to the keyboard, the maximum number of combinations is forty (8 x 5 = 40) keys.

The numbers keys are usually the most used, and are not very convenient keys to use when great accuracy is required. If you use number keys a lot in games or business programs, you might like to build a separate numeric pad. This will enable you to speed up the entry of numbers because you can 'feel' the keys positively hitting the end stop, and thus release it quickly. As the Sinclair keyboard is made out of three thin pieces of plastic film, there is very little distance between the top and the end stop of the key movement (0.1 inch). It is therefore not easy to tell whether you have pushed the key down far enough to make the switch close. The movement of most keyboard "PUSH TO MAKE" switches is at least 0.5 inch, which gives the keys much more positive feel when pressed.

The best type of key switches to use are those with a removable clear plastic top. You can then place a piece of paper under the covers, on

which the keyboard symbols can be written. Eleven keys are required, as the numbers 0-9 are not a lot of use if you cannot RUBOUT any mistakes, except by going back to the Sinclair keyboard. The RUBOUT key• requires the pressing of two keys together, 0 and SHIFT. Therefore the SHIFT key must be included on the numeric pad. Pressing the SHIFT key on its own does nothing, so hitting it accidentally does not give an error on INPUT.

Having the SHIFT key on the numeric pad also means that all the cursor moving keys are also available, SHIFT 5(←), SHIFT 6(↑), SHIFT 7(↓) and SHIFT 8(→). These can be used to quickly EDIT programs, along with the EDIT key which is SHIFT 1. As all of these keys can be reached with one hand if they are grouped in a square, it means the other hand is free to do other things, such as follow a program in a book or a set of data to be INPUT. This can be very useful, as it is easy to lose one's place when trying to watch the screen and the written program at the same time.

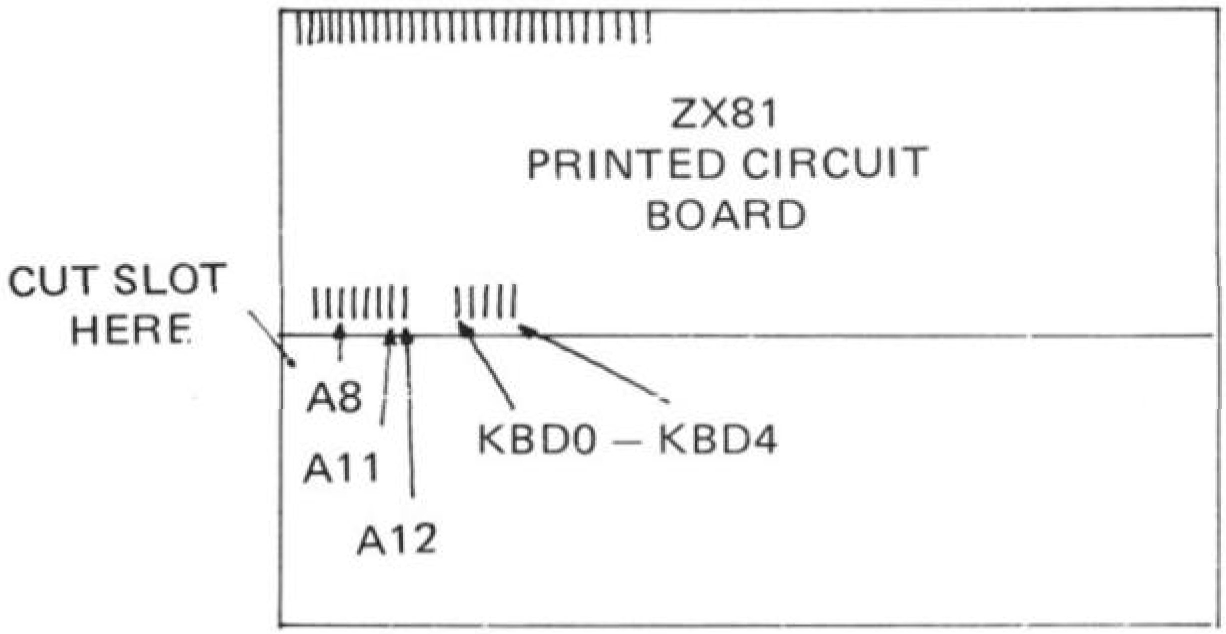
As the keys 1-5, 6-0 and SHIFT are all on different address lines, all three must be included on the numeric keypad. These are A8 (SHIFT), A11 (1-5) and A12(6-0). We also need ALL of the (K)EY(B)OARD-(D)ATA lines (inputs to the computer) KBD0–KBD4.

The keyboard port KBD0-4 is addressed by the ZX81 ROM as INPUT PORT 254 (FE in HEXADECIMAL). BUT because of the way Sinclair addresses his ports, the keyboard port appears at every EVEN INPUT PORT address. That is when address line A0 is at Binary 0, the IOREQ and the WR are Binary 0.

The upper eight address lines (A8-A15) reflect what was in the B register at the time of calling for an input from the port. So the setting of a bit in the "B" register to Binary 0 addresses that key (the address line to 0 volts) and then looks at the result on the data lines. When a key is pressed, the appropriate data line will also be Binary 0.

These actions are all done by the BASIC ROM when using INPUT or INKEY$. This information has only been included for the machine code programmer.

We must open up the casing of the ZX81 to get at the connections on the printed circuit board inside, and thereby

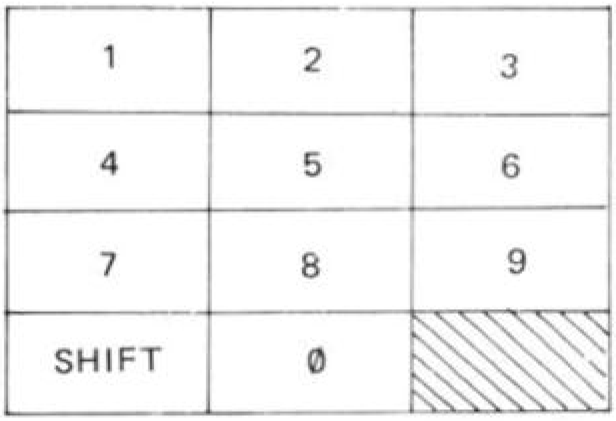



| QUANTITY | COMPONENT |
|---|---|
| 11 | KEYBOARD 'PUSH-TO-MAKE' SWITCHES |
| 8 | PIECES OF WIRE 18 INCHES LONG |
| 1 | BOX |

the data lines.

If you turn the ZX81 upside down, you will see four stuck-on rubber feet. Under three of these feet are screws which need to be removed before the case can be opened. They are under the front two feet and the back left side foot. There are a total of six screws to be removed, ALL of them need to be removed with a small-headed screwdriver, in order not to damage the slot in the screw. Once the screws are taken out, the bottom half of the casing can be removed and the printed circuit board can be seen in the top half, secured by two more cross-cut screws into the top casing. By the bottom left hand side of the printed circuit board you can see the two white plastic strips which connect the Sinclair keyboard to the printed circuit board. These must not be damaged by dropping hot solder on them, so cover them up with a piece of paper. These keyboard strips go into two sockets on the underside of the printed circuit board. The solder strips on the top of the printed circuit board which connect the sockets to the rest of the ZX81 is where we will solder the wires, which we will use to attach the numeric keyboard.

These solder connections consist of a group of eight address strips and a group of five KBD strips. Soldering onto these strips will NOT disconnect any of Sinclair's keyboard functions. None of the wires connecting the ZX81 and the numeric keypad must be over 18 inches long or this causes problems in operating BOTH keyboards. Also make sure that no shorts are made between the strips (see the SOLDERING instructions).




SUGGESTED LAYOUT

A slot must be cut in the left hand side of the bottom casing to lead the wires out. This may be done by making two saw cuts ½ inch apart, ¼ inch deep, with a small hacksaw. Then with a pair of pliers, grip the area between the saw cuts and bend the plastic backwards and forwards until the piece breaks off.

The wiring to the keys, in comparison to the ZX81's, is a piece of cake. The connections are shown in the circuit diagram. The keys have only two tags and these can be connected either way round. The address lines connect five keys and must be wired from key to key, using the wire now attached to the ZX81. There is only one data line (KBD) to each key and only one address line to each key. The SHIFT key only must be wired to address line A8.

The keys can be arranged in any order you like, but a suggested layout is given.

"20 Simple Electronic Projects for the ZX81" by Stephen Adams is published by Interface Publications. Contents of this article ©

# Pig Latin Generator

**Teach your ZX81 to speak 'Pig Latin' with this amusing program by Hans Beerbernon.**

This 1K program uses the ZX81's 'slicing' technique on strings to turn English text, which you enter one word at a time, into 'Pig Latin'. Once you've run it a few times, try to write a 'Pig Latin translator' to decipher the Pig Latin messages given here back into English. Note that line 80 starts a new print line at random, to stop words wrapping around.

```
10 REM PIG-LATIN
20 REM (C) HANS HEERBERNON,
                  MAY, 1982
40 PRINT "ENTER YOUR MESSAGE,"
"WORD BY WORD"
45 PRINT "ENTER $ TO END"
47 PRINT
50 INPUT A$
55 IF A$="$" THEN STOP
60 LET A$=A$(2 TO )+A$(1)+"A "
70 PRINT A$;
80 IF RND>.7 THEN PRINT
90 GOTO 50
```

```
ENTER YOUR MESSAGE,
WORD BY WORD
ENTER $ TO END

XZA OMPUTINGCA SIA
HETA
ESTBA
NDAA IGGESTBA
AGAZINEMA
ORFA
HETA
INCLAIR59
SERUA NDAA OSA AYSA LLAA FOA
SUA
ONTDA EWA

ANYMA
EOPLEPA ELIEVEBA HISTA SIA
AA OREIGNFA
ANGUAGELA UTBA OUYA
NDAA AA
ANMA NOWKA IFFERENTLYDA

NOA
HETA IFTEENTHFA FOA ULYJA
RESIDENTPA
IXONNA
NNOUNCEDAA
EHA
ASWA OINGGA OTA
HINACA SAA
HETA
ESULTRA
FOA NAA
NVITATIONIA
ROMFA
EKINGPA IA MAA OINGGA
EHA AIDSA
NDAA EHA ENTWA

IA
ETBA
NAA BMIA
AIN-FRAMEMA ANTCA
OPECA ITHWA
ROGRAMSPA
FOA UCHSA TUNNINGSA
OMPLEXITYCA SAA HISTA X81ZA NEOA

ODA
OUYA
NDERSTANDUA HATWA
IA
MAA AYINGSA
ROA ODA IA AVEHA OTA ROVEPA TIA
```

# Breaking out

**The first issue of ZX COMPUTING included an article by Toni Baker designed to act as an introduction to machine code. L G Scotford of Eastbourne, East Sussex took up the challenge of developing a BREAKOUT program from the information in Toni's article.**

The program uses the full 24 lines of the screen; line 10 enables this. The main problem was testing to see if the ball was lost. However, eventually the following solution was found:

The very bottom line of the screen is filled with the character used for the bricks. Line 210 then tests to see whether or not the ball is lost by finding the current address of the ball. So, if the variable A is 1, showing that a brick has been hit, and the current address of the ball is greater than the starting position, then the ball is lost. If the position of the ball is less than its starting address then it must have hit a genuine brick, so 1 is added to the score.

The bat is made of three inverse spaces, since the ball will automatically bounce off these. It can be moved right or left by keys 8 and 5 respectively. If all the bricks are cleared then the player is given a bonus ball and a new screen is set up.

The machine code remains unchanged and can be loaded into the REM statement before the BASIC program is entered. The BASIC itself actually slows down the ball to a playable rate without seriously cutting the speed.

The best score I have yet achieved is 136 but no doubt there are many who could easily beat this.

In the article in the last issue, Toni gave a BASIC routine for loading machine code, taken from the book *Mastering Machine Code on the ZX81*.

This is the routine:

```
10  INPUT X
20  LET A$ = "" 
30  IF A$ = "" THEN INPUT A$
40  IF A$ = "S" THEN STOP
50  POKE X,16*CODE A$ +
    CODE A$(2) - 476
60  LET X = X + 1
70  LET A$ = A$(3 TO) note
    there is nothing between
    "TO" and ")"
80  GOTO 30
```

Above this, you need the following REM statement to hold the machine code:

```
1 REM 123456789012345678
      901234567890123456
      789012345678901234
      567890123123456789
      012345678901234567
      890123200
```

Now RUN the program and input the following (counting "/" as "newline"):

```
16516/0101/2A8240/3600/
3A8440/3D/2002/23/23/2B/7E/
FE80/200B/2A8240/4A8440
ED44/328440/228240/3A8540
/3D/2006/11DFFF/19/1804/
112100/19/7E/FE80/200B
2A8240/3A8540/ED44/328540
/010000/7E/FE08/2009/03/
3A8540/ED44/328540/228240
/3634/C9/S
```

## BREAKING OUT

```
 10 POKE 16418,0
 20 LET TS=0
 30 LET B=3
 40 PRINT "32 inverse spaces"
 50 PRINT "inverse space, 30 spaces, inverse space"
 60 same as 50
 70 PRINT "inverse space, 30 graphic H, inverse space"
 80 same as 70
 90 FOR I=1 TO 18
100 same as 50
110 NEXT I
120 LET BP=200
130 same as 70
140 LET S=0
150 LET P=15
160 LET M=PEEK 16396 + 256*PEEK 16397
170 LET X=M+BP
180 POKE 16514,X-256*INT (X/256)
190 POKE 16515, INT (X/256)
200 LET A=USR 16518
210 IF A=1 AND (PEEK 16514+256*PEEK 16515)>X
    THEN GOTO 300
220 IF A=1 THEN LET S=S+1
230 PRINT AT 21,P;"space, 3 inverse spaces, space"
240 LET P=P+(INKEY$="8" AND P<26)-(INKEY$="5"
    AND P>1)
250 IF S=60 THEN GOTO 400
260 GOTO 200
300 FOR I=1 TO 75
310 NEXT I
320 LET TS=TS+S
330 LET B=B-1
340 LET BP=BP+INT (10*RND+1)
350 PRINT AT 21,P;"5 spaces"
360 IF B>0 THEN GOTO 130
370 PRINT AT 8,6;"YOU SCORED ";TS;" POINTS"
380 STOP
400 FOR I=1 TO 10
410 FOR J=1 TO 5
420 NEXT J
430 PRINT AT 10,10;"BONUS BALL" (underlined)
440 FOR J=1 TO 5
450 NEXT J
460 PRINT AT 10,10;"BONUS BALL"
470 NEXT I
480 LET TS=TS+S
490 LET B=B+1
500 CLS
510 GOTO 40
```

## WORDSQUARE

This program is of the "wordsearch" variety and will fit a list of words onto a grid whose dimensions depend upon the length of the longest word in the list. It is written specifically for the ZX81 and makes extensive use of the "print at" statement. This means it would require a lot of modification to run on another system. It needs about 4K.

The program has been designed in modules in an attempt to make it easy to understand and modify the flow.

*Lines 10 to 260* are the initialization process. The words which are to be used are stored in the string array C$. The longest word must be input first so that the size of the array can be determined. A check is made in line 170 to make sure that none of the words are too long for the array. If this is the case then the word is not accepted and a new word must be input.

*Lines 200 to 260* print the wordsquare grid onto the screen.

*Lines 270 to 550* are the main part of the program and actually fit the words into the square. A 2 dimensional array is first set up to store the co-ordinates finally chosen for the characters in each word (H$). The current word is assigned to variable J$ and random starting co-ordinates (X and Y) and displacements (Z and W) are chosen in lines 310 and 370.

*Lines 390 to 480* single step through the word, fitting each character into the square and storing its co-ordinates *temporarily* in the 2 dimensional array K. If the word runs off the square when the co-ordinates are incremented by the displacement, or the chosen co-ordinates are already filled by an unsuitable letter from another word, the current word is started again with new X, Y, Z and W variables. Only when the current word has been completely fitted in will its characters be entered in the final array and be printed to the screen by lines 490 to 540.

*Lines 560 to 650* fill all the vacant spaces on the grid with random letters. If you do not wish to see the words as they are fitted into the grid, you can specify this at the start. The program will then only print in the words as it generates the random letters.

*Lines 700 to 750* will show you the positions of the words when you get bored looking for them by inversing them on the square when requested to do so.

There is also a visual indication of the progress made on each word as the program is running.

**Variables used**

i) Simple numerical variables

- A — number of words in the list.
- D — size of the square (length of longest word plus 2)
- X — X coordinate
- Y — Y coordinate
- Z — displacement to X coordinate
- W — displacement to Y coordinate

ii) Simple string variables

- B$ — longest word
- D$ — current word input
- J$ — current word in square
- P$ — random letter
- R$ — set for secret generation of square
- Q$ — set for printing of answers

iii) Numerical arrays

- K — temporary store of coordinates

iv) String arrays

- C$ — list of words
- H$ — store for final positions for each letter

All other variables are the control variables for loops involved in input of word lists, printing to the screen or arrays or character fitting.

The longest word in the list should have no more than 18 letters or the grid will not fit on to the screen. About 20 words of varying length can be fitted in about 5-10 minutes. A longer list of words can result in a very frustrating wait.

It is a good idea to enter the words in descending order of length as this will speed up operation. The program is fascinating to watch in operation, so run it in SLOW.

```
   1 REM WORDSQUARE
   2 REM BY J ELLIOTT
  10 PRINT "IF YOU DO NOT WISH TO SEE"
  20 PRINT "THE ANSWERS THEN ENTER ""N"""
  30 PRINT "NOW. OTHERWISE PRESS ANY KEY"
  40 LET R$=INKEY$
  50 IF R$="" THEN GOTO 40
  60 CLS
  70 PRINT AT 0,10;"WORDSQUARE"
  80 PRINT AT 19,0;"HOW MANY WORDS"
  90 INPUT A
 100 PRINT AT 19,0;"ENTER LONGEST WORD"
 110 INPUT B$
 120 DIM C$(A,LEN B$)
 130 LET C$(1)=B$
 140 FOR C=2 TO A
 150 PRINT AT 19,0;"ENTER WORD NUMBER ";C
 160 INPUT D$
 170 IF LEN D$>LEN B$ THEN GOTO 150
 180 LET C$(C)=D$
 190 NEXT C
 199 REM NEXT LINE CONTAINS 22
         SPACES
 200 PRINT AT 19,0;"                      "
 210 LET D=LEN B$+2
 220 FOR E=1 TO D
 230 FOR F=1 TO D
 240 PRINT AT E,F;"*"
 250 NEXT F
 260 NEXT E
 270 DIM H$(D,D)
 280 FOR Q=1 TO A
 290 LET J$=C$(Q)
 300 PRINT AT 19,0;J$
 310 LET X=INT (RND*D)+1
 320 LET Y=INT (RND*D)+1
 330 LET Z=INT (RND*3)
 340 LET W=INT (RND*3)
 350 IF Z=0 AND W=0 THEN GOTO 330
 360 IF Z=2 THEN LET Z=-1
 370 IF W=2 THEN LET W=-1
 380 DIM K(LEN J$,2)
 390 FOR L=1 TO LEN J$
 395 REM    SINGLE SPACE IN QUOTE MARKS IN NEXT LINE
 400 IF J$(L)=" " THEN GOTO 480
 410 LET X=X+Z
 420 LET Y=Y+W
 430 IF X<1 OR X>D OR Y<1 OR Y>D THEN GOTO 290
 435 REM    SINGLE SPACE IN QUOTE MARKS IN NEXT LINE
 440 IF (NOT H$(X,Y)=" ") AND (NOT (H$(X,Y)=J$(L))) THEN GOTO 290
 450 LET K(L,1)=X
 460 LET K(L,2)=Y
 470 PRINT AT 19,L-1;CHR$ (CODE J$(L)+128)
 480 NEXT L
 490 FOR M=1 TO LEN J$
 495 REM    SINGLE SPACE IN QUOTE MARKS IN NEXT LINE
 500 IF J$(M)=" " THEN GOTO 540
 510 LET H$(K(M,1),K(M,2))=J$(M)
 520 IF R$="N" THEN GOTO 540
 530 PRINT AT K(M,1),K(M,2);J$(M)
 540 NEXT M
 550 NEXT Q
 555 REM 15 SPACES IN NEXT LINE
 560 PRINT AT 19,0;"               "
 570 FOR N=1 TO D
 580 FOR P=1 TO D
 585 REM    SINGLE SPACE IN QUOTE MARKS IN NEXT LINE
 590 IF NOT H$(N,P)=" " THEN GOTO 630
 600 LET P$=CHR$ (INT (RND*26)+38)
 610 PRINT AT N,P;P$
 620 GOTO 640
 630 PRINT AT N,P;H$(N,P)
 640 NEXT P
 650 NEXT N
 660 PRINT AT 19,10;"FINISHED"
 670 PRINT AT 20,0;"PRESS ANY KEY FOR ANSWERS"
 680 LET Q$=INKEY$
 690 IF Q$="" THEN GOTO 680
 700 FOR N=1 TO D
 710 FOR P=1 TO D
 715 REM    SINGLE SPACE IN QUOTE MARKS IN NEXT LINE
 720 IF H$(N,P)=" " THEN GOTO 740
 730 PRINT AT N,P;CHR$ (CODE H$(N,P)+128)
 740 NEXT P
 750 NEXT N
```

# ZX81 GAMES

*from*

## J.K. GREYE SOFTWARE

THE NEW GENERATION SOFTWARE HOUSE

*"Without question the finest machine code games available today".........* J.N. ROWLAND Product Manager for W.H. SMITH.

**GAMESTAPE 1 for 1K** ________ **only £3.95**
10 Games incl. ASTEROIDS, UFO, CODE, BOMBER, GUILLOTINE, KALEIDESCOPE, etc.
**PROBABLY THE BEST VALUE 1k TAPE AVAILABLE.**

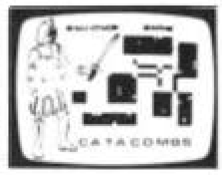


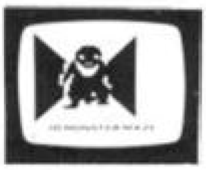

We've done in 1k games which some of our competitors required 16k to do!




**GAMESTAPE 2 for 16K** ________ **only £3.95**
***STARFIGHTER** Superb machine code Space Battle. Set against a background of twinkling stars, with stunning explosions — if you can hit the enemy!
**PYRAMID** Can you move the Pyramid? Make a mistake and it will colla...! A Thinkers game.
**ARTIST** The ultimate Graphic Designers aid. 8 Directions, 10 Memories, SAVE, COPY, RUBOUT, CLS, etc.

**GAMESTAPE 3 for 16K** ________ **only £4.95**
***CATACOMBS** A Multi-Level Graphics Adventure. Each level can contain up to 9 Rooms, 8 Passages, 7 Monsters, Food, Gold, Traps, Phantoms, an Exit (to the next level), and there's an infinite number of levels.
NOTE. . . . This is NOT one of the necessarily limited text Adventures as sold elsewhere.
*"An excellent addictive game which will keep you amused for hours." . . . . COMPUTER & VIDEO GAMES.*



**GAMESTAPE 4 for 16K** ________ **only £4.95**
***3D MONSTER MAZE** The Game to Top All Others. Unbelievable Graphics! Can you find your way through the Maze? The EXIT is there somewhere, but then so is a T.REX, and its after YOU! All in 3D (the T.REX will actually run towards you in full perspective!), you've never seen anything like this before!
*"3D MONSTER MAZE is the best game I have seen for the ZX81"* . . . . COMPUTER & VIDEO GAMES
*"If I had to choose just one programme to impress an audience with the capabilities of the ZX81, then J.K. Greye's 3D MONSTER MAZE would be the one without doubt"* . . . . ZX COMPUTING.

**GAMESTAPE 5 for 16K** ________ **only £4.95**
***3D DEFENDER** The Ultimate Space Game. Super fast Machine Code 3D version of the Arcade favourite. You have to save your home planet from the marauding Alien Spacecraft. This is all in 3D, your viewscreen shows you the view out of your fighters cockpit window. The backdrop moves when you turn, or fly up or down (8 flight directions), just as if you were really flying it! But then YOU ARE! The Enemy Saucers will actually zoom towards you in 3D, and shoot you if you let them! Your display includes Score, Shield Strength, Altitude, Proximity, Forward Radar and your viewscreen, which shows your rotating home planet, backdrop of Stars, Meteors, Explosions, Plasma Blasts, your Photon Beams, up to 4 Enemy Saucers and of course its all in full 3D!
**A SMASH HIT at the ZX Microfair (most of the other software houses wanted a copy), a game not to be missed!**






**GAMESTAPE 6 for 1K** ________ **only £1.95**
***BREAKOUT** Super Fast Full Screen Display Game. Your all time favourite with an added twist. See how much Money you can win and watch the pounds convert to Dollars. All in Machine Code for Fast Action with 3 Speeds, 2 Bat Sizes and three angles of rebound! The best BREAKOUT around and at this price you can't go wrong!

GAMES MARKED * INCL. MACHINE CODE.
Prices include VAT and U.K. P. & P.
(Add appropriate Postage on Foreign Orders). Cheques/P.O.s to

**J.K. GREYE SOFTWARE**
Dept.ZX, 16 Park St., Bath, Avon BA1 2TE.
CREDIT CARD SALES: Phone: 01-930-9232 (9 a.m. - 7 p.m.)
FOR INSTANT DESPATCH

If you prefer to see before buying, our range of **GAMESTAPES** are stocked by the following stores.

**BUFFER MICROSHOP** 374A, Streatham High Rd., London SW16
**GEORGES** 89 Park St., Bristol, Avon.
**MICROSTYLE** 29 Belvedere, Lansdown Rd., Bath, Avon.
**MICROWARE** 131 Melton Road, Leicester.
**SCREEN SCENE** 144 St. Georges Rd., Cheltenham, Glos.
**W.H. SMITH** Computer Branches.
**ZEDXTRA** 5 School Lane, Kinson, Bournemouth, Dorset.

TRADE & EXPORT ENQUIRIES WELCOME

# ZX81 M.C. 16K SOFTWARE

## ZXC ARCADE PLUS PACK:

SLOT — A graphical simulation of a fruit machine. With Hold and Nudge features.
BREAKOUT + SUPER PROGRESSIVE BREAKOUT
DODGEMS — with increasing difficulty. Can you evade the computer cars and clear all the dots.
LIFE — On a maximum 64x48 screen grid
Send cheque, postal order for £7.95 to:

## A. READMAN SOFTWARE,

16 South Row, Eldon, Biship Auckland, Go. Durham
DL14 8UT
Mail order only, trade enquiries welcome

# WRITE EFFICIENT ZX81 BASIC

**— Improve your ZX BASIC programming skills with this new book.**

**— Assumes knowledge of the Sinclair ZX81 BASIC manual only.**

**— Covers many techniques for speed, space saving and "good practice".**

**— Illustrated by over 25 useful and enjoyable programs, demonstrating the rules described, while making the most effective use of 1K.**

**— Many of the principles listed are of more general application, most particularly to ZX SPECTRUM Basic.**

**— Send £3 to:**

**Ivor Killerbite,
10, Elson Road,
Formby,
Liverpool L37 2EG**

# RAM, sweet RAM

**In this review, Tim Langdell from West Dulwich looks at a number of RAM packs and assesses their value for the ZX81 owner.**

There's no special reason for our choosing these particular add-on memories, and they differ quite a lot in their price (from £20 to about £55) and in their facilities. They are aimed at different markets to some extent, and should not be directly compared without bearing this in mind. Hopefully, though, by discussing each of their features and quirks I may be able to give you some idea of what you might be looking for in a RAM pack.

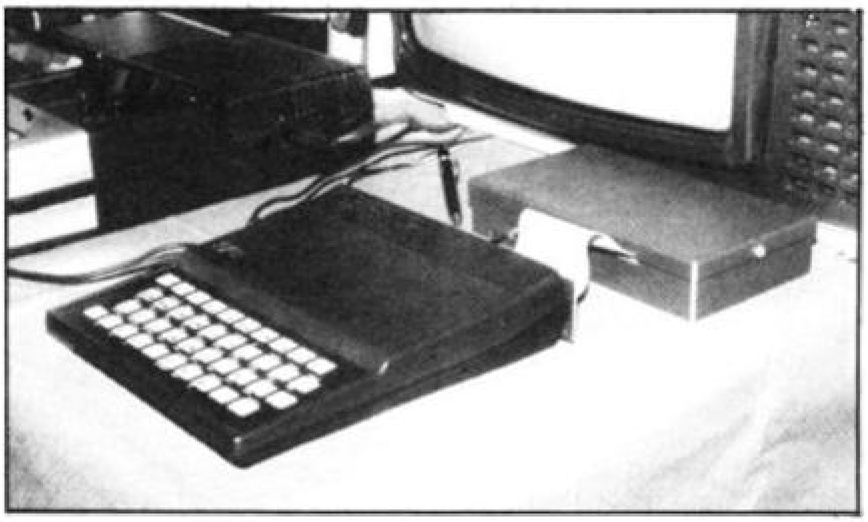
The main five RAM packs being considered are all 16K byte ones, although two of them have bigger 56K byte sisters (and I will mention these too). I'll also be comparing each of them with Sinclair's own 16K RAM pack to give you some idea of how they differ. The RAMs are from Byg Byte, Taurus, Downsway, Memotech, and Plessis.

First, the original Sinclair memory pack. When this first came out it was the only one you could buy for your Sinclair computer, and at that time it was a compact, reasonably-priced design. However, most people who bought one of these RAM packs noticed that it buzzed when in use and did not like being moved or used for too long. Some of the Sinclair packs overheated quickly leading to a 'crash', and some needed only to be moved a fraction for the memory to be lost and the now infamous 'white out' to occur. Some people helped these problems by taking the RAM out of its case and using vaseline on the rear connector but these modifications seemed a bit unnecessary.

By the end of last year, RAM packs had hit the market which were more reliable than Sinclair's and didn't buzz. By the beginning of this year these RAM packs were also selling for some £10 less than the Sinclair, which more realistically reflected the drop in the cost of electronic components over the last two years.

Now there are at least a dozen 16K RAM packs for the ZX81 on the market, and knowing which one to chose can be difficult. A price war seems to have started over the past three months to see who can sell a 16K RAM for least. One of the earlier RAM packs to be launched was Downsway's 16K one, although it was sold through Hilderbay, Buffer Micro Shop, and JRS. Now Downsway distribute RAM packs themselves and have added a 56K RAM to their range.

Their 16K RAM pack is very small and light. It comes in a black plastic box with a gold plated edge connector protruding from the bottom. It is simple to slot into the rear of the ZX81 in the same way as the Sinclair RAM. However, it is lighter than the Sinclair and has a foam strip across it which reduces wobble (and hence potential crashes) to a minimum. Unlike the Sinclair (but like all the other RAMs in this review) Downsway's RAM did not buzz when in use. In fact, my main complaint (if not only) about it was that it had no duplicate edge connector at the rear of it, which means that it must either be the only add-on at the back, or at least the last to be added on. This is true of the Sinclair RAM pack too, of course, and many others as well.



The Byg Byte RAM pack is a newer addition and true to its name is one of the biggest 16K RAMs on the market. Byg Byte claim that they put it in a large black plastic box to improve its stability. But this means that it is about three times the size of the Downsway RAM and not really any less stable, and I have my doubts about their reasoning. Nonetheless, it was also a reasonably stable RAM pack which ran happily for hours. Unlike other RAMs it has a power-on light-emitting diode which lights up to tell you the RAM is switched on. I must say that I did not see much point in this feature other than to remind you whether the whole computer is on or off. Should you be curious, 90 per cent of the inside of the Byg Byte is empty space . . .

Memotech pioneered the big memory scene for the ZX81 when they brought out their 48K RAM extension last year. Even at about £125 this sold well and paved the way for the recent 56K byte RAMs which have suddenly appeared. Clive Sinclair envisaged ZX81 users adding no more than 16K bytes of RAM to their machine and at first it was said to be impossible to add more than this. The reason for this was that the ZX81 uses the address line 15 (A15) to produce the TV display and yet a logical high on this address line is what the ZX81 needs to detect when addressing memory space above the 32K mark. For the technical buffs the solution to the dilemma lies in realising that when the M1 line is low and A15 is high, a display is being generated, but if A15 is high at other times it must be because the line holds a valid address. No one, then, was too suprised to see Memotech produce another first with a 56K byte RAM early this year, but it was quite recently that they started offering a 16K RAM pack too.

The new RAMs from Memotech are beautifully designed and blend in really well with the styling of the ZX81. They come in black anodised aluminium cases shaped to the contour of the rear of the ZX81, and therefore fit like a glove. There is virtually no wobble, but chances of wobble can be further reduced by using the foam strip which Memotech supply. Unlike almost all other RAM packs, the Memotechs have a duplicate rear connector coming out the back. It is thus quite easy to add more hardware on. Like all the other 16K RAM packs (with the exception of the Taurus) the Memotech one uses 4116 industry standard RAM chips and the whole assembly has a very professional feel to it. The 4116 RAM chips are each 16K bytes by one data line, and so eight are needed for a 16K RAM. But these chips whilst popular are not best suited for the latest micros. They need not only the usual 5 volts to run them but also a −5 V and 12 V supply. Makers had thus been waiting for the new 64K-by-

one chips to come down to a reasonable price for these only need a single 5 V supply, draw very little current, and you need only eight of them for a potential 64K bytes of memory!

Memotech were the first to use these new 4564 chips, soon followed by others such as Downsway. Both the Memotech and Downsway larger memory packs are almost identical to look at as their 16K counterparts. The major difference between the Memotech packs is that the so called 64K version has four switches visible in its rear which allow you to switch out the area between 8K and 16K in the memory map in 4K blocks. This is an excellent idea and I hope other manufacturers will follow this lead.

The microprocessor in the ZX81 (the Z80A) can only address 64K of memory and the first 8K of this is taken up with Sinclair's ROM (ReadOnly Memory) containing the software to run the machine, provided the BASIC and so on. Therefore the very biggest add on memory can only be 56K, and it was rather misleading of Memotech to refer to it as a full 64K RAM pack. To confuse matters, many other manufactuers copied Memotech's use of the term '64K RAM pack' just in case you though the Memotech one was bigger! Both the Memotech and Downsway 56K packs can be obtained by part exchanging your 16K RAM. Memotech give you three months to return your 16K Memopak for an upgrade, whereas Downsway seem happy to consider any 16K RAM in working order in part exchange. In use the big RAM packs are identical; giving 16K or RAM for BASIC programs, the top 32K area where you can store data, arrays, etc, and the 8K space between 8K and 16K where machine code can be run, or programs or data can be stored for transference between programs (this area of memory remains intact after NEW or after loading another program). It is important to remember, though, that with one of these bigger memories you have mapped all the available memory space leaving none for other add-ons to use such as character generators, sound boards, or memory mapped I/O ports. Only the Memotech allows you to add something between 8K and 16K, but many add-ons are mapped in the 32K region and are thus not usable.

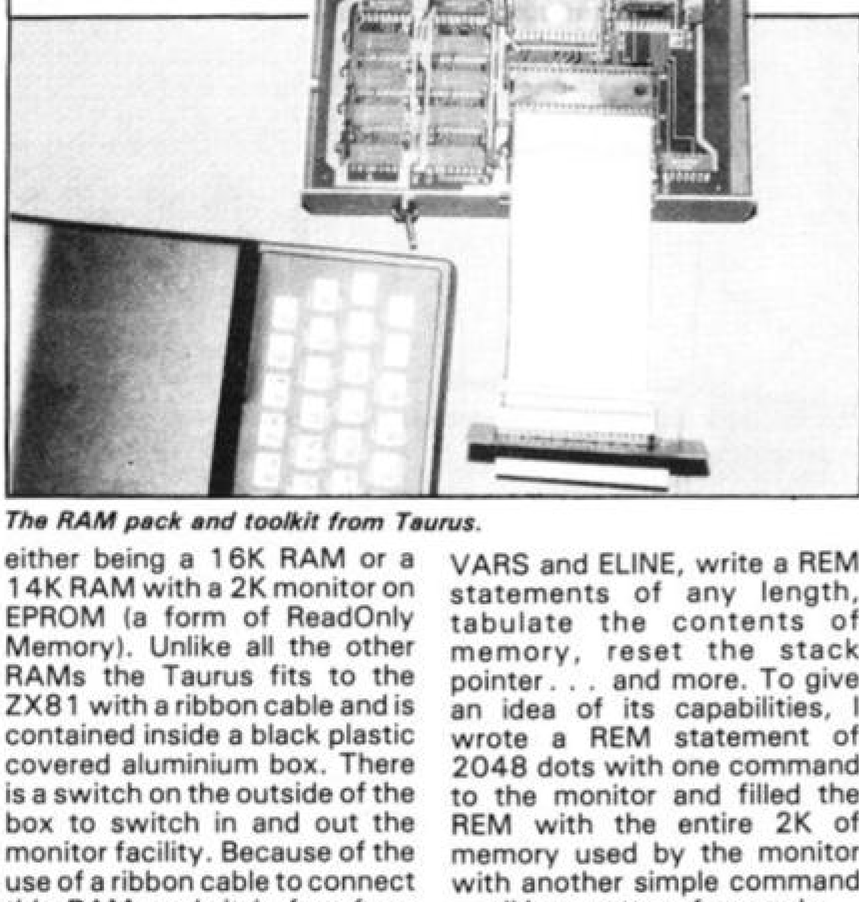
The RAM pack from Taurus offers the unique facility of either being a 16K RAM or a 14K RAM with a 2K monitor on EPROM (a form of ReadOnly Memory). Unlike all the other RAMs the Taurus fits to the ZX81 with a ribbon cable and is contained inside a black plastic covered aluminium box. There is a switch on the outside of the box to switch in and out the monitor facility. Because of the use of a ribbon cable to connect this RAM pack it is free from any problems of poor contact and wobbling. If you are interested in serious programming using machine code, then you may find the Taurus monitor very useful. I found some of its capabilities almost awesome, and nearly all of them very useful. Briefly, the monitor allows you to do hexadecimal arithmetic, set/clear/ or display breakpoints, copy data from one area of memory to another, do decimal to hex conversion and vice versa, fill an area of memory with a constant, move the contents of one area of memory to another, read/ write a port, display the state of the registers, display the contents of DFILE DFCC VARS and ELINE, write a REM statements of any length, tabulate the contents of memory, reset the stack pointer . . . and more. To give an idea of its capabilities, I wrote a REM statement of 2048 dots with one command to the monitor and filled the REM with the entire 2K of memory used by the monitor with another simple command — all in a matter of seconds.

The usual method of typing in 2048 characters and running a FOR/NEXT loop to load the data into memory seems ridicously slow by comparison. The RAM pack part of the Taurus uses the less well known 2118 low power RAM chips. The RAM performed perfectly well, but I did have a reservation about the availablity of the chips should anything ever go wrong.

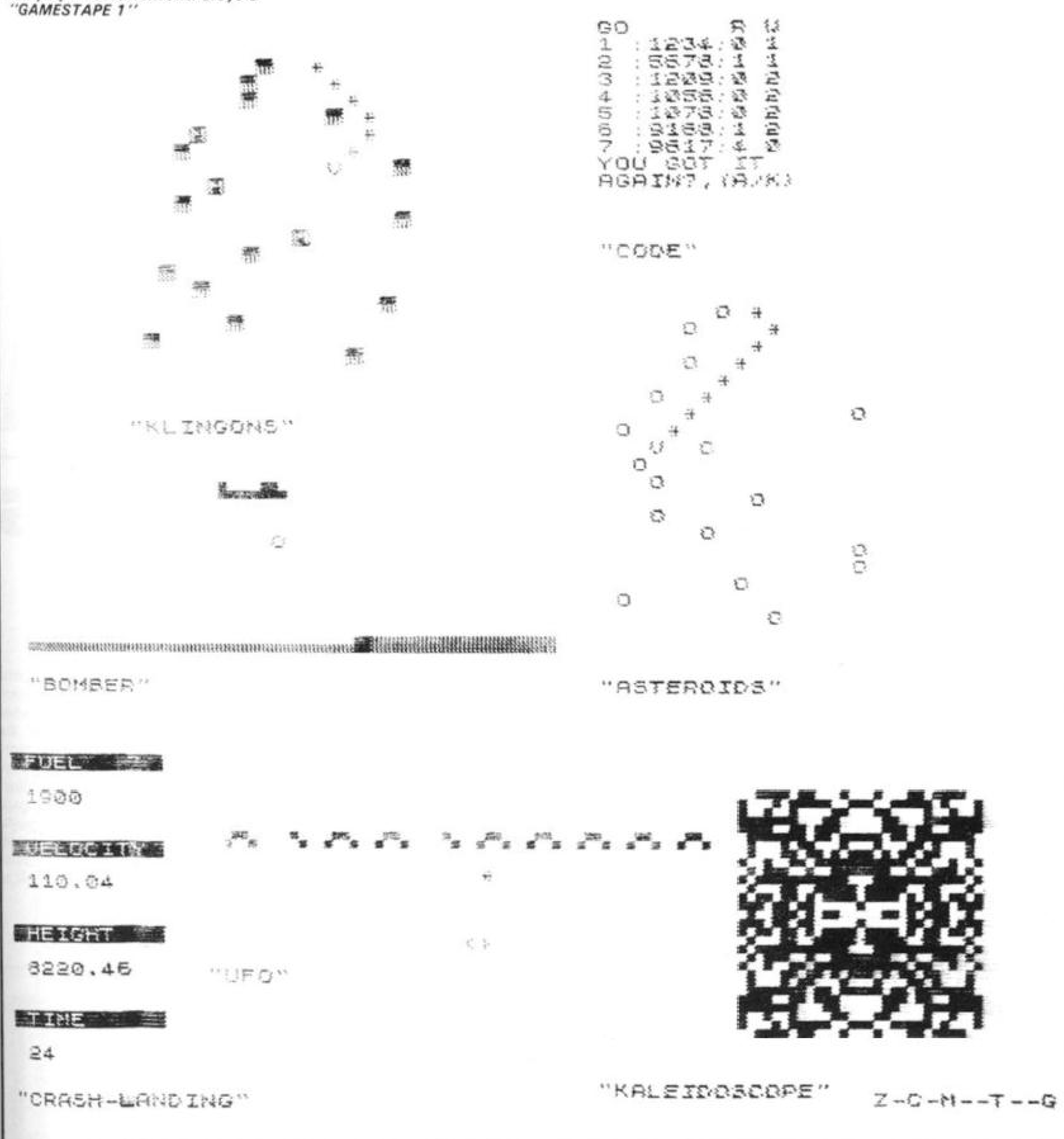
Finally, a new RAM pack has just come into the market and is manufactured by Plessis Electronics. It has been introduced at the very low price of £19.95, undercutting the cheapest other RAMs by up to £10. It comes in a black plastic box similar to Byg Byte's, but about half the size. Like the other RAM packs it has a gold plated edge connector, and does not buzz when being used. Plessis seem to have succeeded in producing a reliable 'no frills' RAM pack which works well and is at a rock bottom price (in fact one wonders how they are making a profit). I would have preferred to see Plessis use a foam strip to reduce the chance of wobble, but that being said I had no problems with programs crashing either due to wobble or overheating.

In terms of value for money the Plessis must take the prize, although by the time this review appears no doubt other RAMs will have been brought down to about the £20 to £25 mark. The Byg Byte worked well but was rather large than it needed to be and at around £30 had little to recommend it over the Plessis at about £20. Sinclair have just brought the price of their RAM pack down to about £30 too, and are no doubt partly responsible for the low price trend which began around the time of the Spectrum's launch. But even reduced in price there is little to recommend the Sinclair offering either for it is still more likely to overheat than the others and makes an annoying buzzing sound. The Downsway 16K RAM is a very neat, small one and fits very firmly onto the ZX81. It is still worth considering at around £25 as a strong competitor to the Plessis, and may of course be cheaper soon. Their 56K RAM pack at about £60 (£47.50 with a 16K pack traded-in) is good value and




*The RAM pack and toolkit from Taurus.*

*Sample printouts from J.K. Greye's "GAMESTAPE 1"*




Bulls' to pre-computer veterans) where the player has to guess a four-digit number. They've done well to fit it into 1K, and I found it quite hard since my usual strategy doesn't apply when duplicated digits are now allowed.

CRASH LANDING is a standard lunar lander program, and I was disappointed to find that, having crashed, I wasn't told what size crater I had made!

Of the graphics games, KLINGONS and ASTEROIDS are pretty much the same program: You move your ship left or right, and the opposition scroll steadily up towards you. The difference (!) is that you have to hit the Klingons, but miss the Asteroids. UFO has another twist. The flying saucers stay put, while your laser base speeds automatically from left to right, and you have to judge the correct moment to fire a missile. Similarly with BOMBER. You have to decide when to release the payload to hit the dam, and your bombing runs gets shorter as your aim improves.

ARTIST is a standard plotting program, allowing you to draw with pixels directed by the cursor controls. A COPY statement is built in, so it is possible to copy the screen onto the printer without stopping the program. KALEIDOSCOPE produces a random symmetrical pattern with pixels winking on and off. Hypnotic stuff.

The last program on the tape is GUILLOTINE, which is a two-player hangman game with a guillotine instead of gallows. You can guess what happens if the player loses. Thank goodness it is not in colour. I was very impressed with the



*The team from Macronics at a recent computer show. They are (from left) Ken Macdonald, Ron Bissell, James Steventon and Jonathan Cranston.*

amount which had been packed into 1K on this one.

Clear and simple instructions are given for each game, and I had no problems with loading. I found no errors in any of the programs, and they included 'data validation' routines where possible. For example, in ARTIST you couldn't plot off the screen, and similarly you couldn't go off the edge in KLINGONS and ASTEROIDS. I would hope that most new ZX81 owners would soon be writing their own Lunar Lander and Mastermind games, but if they are in a hurry to see what their machine is capable of, then this tape provides ten good examples.

# Here come de Galaxians

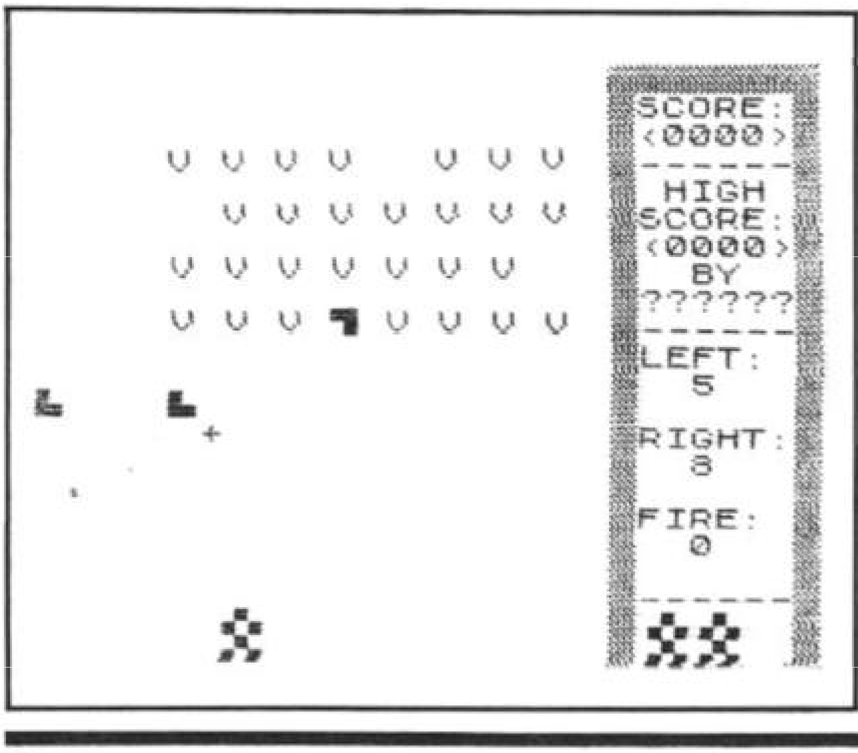
## Always daring to brave the dangers of deepest ZX Space, Jim Robart takes on the might of Artic's Galaxians.

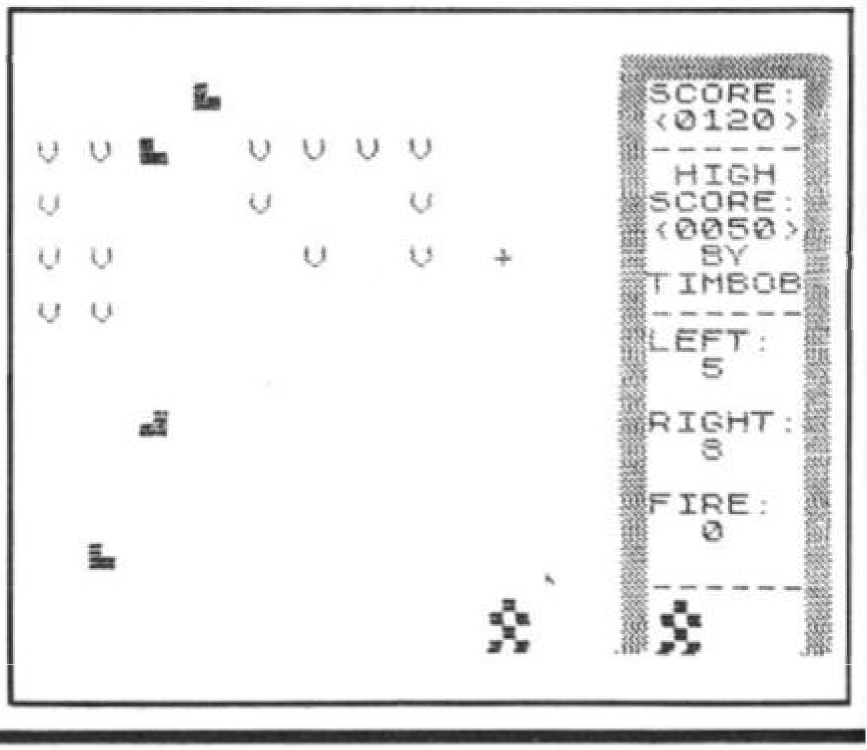
My first reaction to the title page, was "Wow". The ZX GALAXIANS opening frame is a stunner. My first reaction to the appearance of the program when running, a mob of the letter V hovering ahead, a vaguely man-shaped thing (the 'spacecraft') built of standard graphic symbols underneath, was a disappointment.

I had not reckoned on the intelligence of the Galaxians themselves. "This is simple," I thought, and proceeded to be wiped out with a miserable score of 20, for hitting one, and one only swooping Galaxian. I decided to concentrate a little more, and after five games, had managed to score as 'high' as 90. I soon learned that swooping Galaxians were to be feared, and they seemed almost impossible to avoid. From time to time my program crashed, if I was holding down a key when a new man was made available, but I imagine (hope) this was a quirk of my particular tape, and not a general fault in the program.

ZX GALAXIANS runs entirely in machine code, and needs 4K. The program listing consists of a screen-long REM statement, a SAVE line, and a RAND USR line. The program runs itself after taking about a minute and a half to load. If it crashes, GOTO 20 will get it running again. The title page is deleted by touching any key. The "5" key moves you left, "8" moves you right and you fire by touching the "0". Points are scored by hitting Galaxians in formation (10) or when swooping (20). Despite the graphics, which are more Sinclair's fault than Artic's, this program — written by William J. Wray — provides a good emulation of the arcade game. If your nerves can stand an attack of swooping graphics symbols, buy it for yourself as an early Christmas gift.

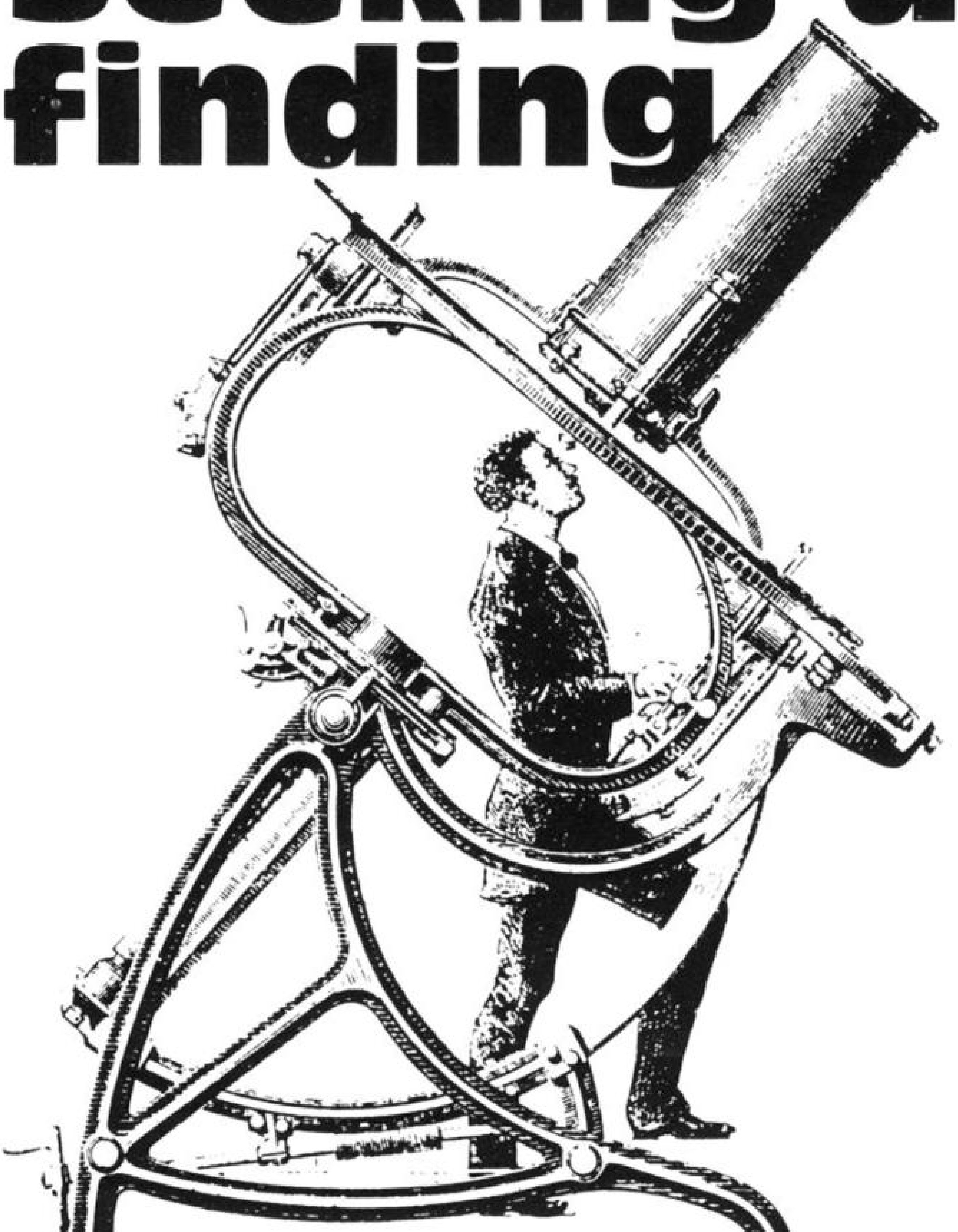
*Artic's Galaxians swoop:*



**ZX80 owners find that there is still a lot to be learned about their machines even after having them for a year or two. Some ZX80 owners have even formed a society to preserve their machines from the onslaught of new ZX computers from Uncle Clive. J. Calderwood of Ballymoney and Fred White from Borrowash, are two ZX80 owners who believe the computer is far from past its peak. And they've sent us these splendid programs to prove it.**

# Seeking and finding



J.R. Calderwood challenges two players to this game, which uses a 1K ZX80. The object of "Seek" is to occupy the same position on the playing area as your opponent. The player moving onto his opponent's position is awarded points depending where on the playing area they are. For example the top of the area is less valuable than the bottom. The right is slightly more valuable than the left.

The two playing positions are set up at random in lines 55 to 60. This position is printed out in lines 260 to 325. The players are shown at this time inside a 10x10 grid and can move around using keys 5,6, 7 and 8. Movement being in the direction of the arrows printed over these numbers. After each key is pressed NEWLINE must be pressed. The number of squares a player can move at a time is limited to a maximum of 10. In fact it will almost always be less than this because of the effect of line 130. This line reads the value of position 16414 and if greater than 200 moves out of the loop allowing no more input during that turn. As this register is incremented 50 times a second and works in modulo 256 it can be seen that during any five seconds there are just four seconds during which input will be accepted. It is surprisingly difficult to judge when inputs will again be accepted. An input of 0 will end the turn.

Although the playing area is shown as a 10x10 square it is possible to move around outside this area. The computer will keep track of your movements but will not print your position, you will need to remember it for yourself! This gives you the opportunity to attack your opponent from hiding.




```
5    PRINT "SEEK"
10   DIM P(2)
20   DIM Z(10)
30   DIM A(2)
40   LET B = 0
50   RANDOMISE
55   FOR I = 1 to 2
60   LET A(I) = RND(195)
63   IF A(I) \ 6 THEN GOTO 60
65   IF NOT ((A(I) - 6)/10= (((A(I) - 6)/10)/2)*2 THEN
     GOTO 60
70   IF A(1) = A(2) THEN GOTO 60
80   NEXT I
90   GOTO 260
115  FOR I = 1 TO 10
116  LET Z(I) = 0
117  NEXT I
118  FOR I = 1 TO 10
120  INPUT Z(I)
130  IF PEEK(16414) / 200 THEN GOTO 150
135  IF Z(I) = 0 THEN GOTO 150
140  NEXT I
150  FOR I = 1 TO 10
180  IF Z(I) = 5 THEN LET A(B) = A(B) - 1
185  IF Z(I) = 8 THEN LET A(B) = A(B) + 20
190  IF Z(I) = 7 THEN LET A(B) = A(B) - 20
200  IF Z(I) = 6 THEN LET A(B) = A(B) + 20
210  NEXT I
220  IF NOT A(1) = A(2) THEN GOTO 250
230  LET P(B) = P(B) + A(2)
235  CLS
240  GOTO 55
250  CLS
260  PRINT ".........."
265  PRINT ".";
270  FOR X = 1 TO 10
275  FOR Y = 1 TO 10
280  LET Z = (20*(X - 1)) + Y + 5
285  IF NOT A(1) = Z AND NOT A(2) = Z THEN PRINT" ";
290  IF A(1) = Z THEN PRINT CHR$(148);
300  IF A(2) = Z THEN PRINT "*";
310  NEXT Y
315  PRINT "."
316  PRINT ".";
320  NEXT X
325  PRINT ".........."
330  PRINT "SCORE";CHR$(148);P(1);"*";P(2)
350  LET B = B + 1
360  IF B / 2 THEN LET B = 1
370  GOTO 100
```

### Mars Landing

From Borrowash, Derby, Fred Whittle sends us this program which also fits within the 1K on a ZX80.

The screen shows the spaceship landing with a magnification effect when you drop below 600 feet. You should enter units of burn up to six. Inputs of seven or above are used to abort the landing, if you lose your nerve. An input of 0 will give you one unit of free fall. The units of burn are calculated in terms of acceleration so an input of, say, four is equivalent to four twos, of 16 ones, hence high inputs save fuel, but beware of trying to save fuel, and entering so much you crash.

With experience, the 60 units of fuel (line 46) will prove more than adequate to both land and take off again. Yes, you do have to take off again.

The program will not allow you to input more burn than you have time left, use more fuel than you've got, or free fall while landed. You leave orbit by reaching a height of 15000. A warning is given when you are running out of fuel. You can only land once. If you want to be able to land more than once, delete (AND NOT F equals 2) in line 180, but this will also mean you cannot enter one to both take off and land again in the same move. Therefore, the only effect is to reduce fuel.

```
20   PRINT "MARS LANDING GAME"
25   PRINT "TO LAND REDUCE BOTH SPEED AND HEIGHT
     TO \ 20"
30   PRINT "HIT NEWLINE"
35   INPUT E$
40   CLS
42   LET A=10000
44   LET B=500
46   LET C=60
50   LET F=1
60   LET Y=18-A/600
65   IF A\600 THEN LET Y=17 - A/35
70   IF A/15000 THEN GO TO 215
75   FOR X=1 TO Y
80   PRINT
85   NEXT X
90   PRINT "          ",;"HEIGHT ";A
95   PRINT "          ",;"SPEED";B
100  PRINT "          ",;"FUEL ";C
105  PRINT "          ",,;"INPUT BURN \7"
115  INPUT D
120  CLS
125  IF D/6 THEN GO TO 225
130  IF D=0 AND A/0 THEN GO TO 305
135  IF D=* AND F=2 THEN GO TO 75
140  IF 2*D/C THEN GO TO 320
145  IF B=0 THEN LET B=1
150  IF D/(2*A)/ABS(B) AND NOT F=2 THEN GO TO 195
155  LET C=C - 2*D
160  IF C\0 THEN GO TO 320
165  LET D=D**2
170  LET B=B - 15*D
175  LET A=A - B*D
180  IF A\20 AND B\20 AND NOT F=2 THEN GO TO 275
185  IF A\0 THEN GO TO 295
190  IF C\20 THEN PRINT "FUEL LOW"
192  GO TO 60
195  PRINT "NOT ENOUGH TIME"
200  GO TO 75
215  CLS
220  PRINT "YOU HAVE LEFT MARS ORBIT"
225  PRINT "ANOTHER GAME?"
230  INPUT G$
235  IF G$="YES" THEN GO TO 40
245  PRINT "END OF GAME"
250  STOP
275  PRINT "LANDED - NOW LEAVE"
280  LET F=2
284  LET A=0
286  LET B=0
290  GO TO 75
295  PRINT "CRASHED"
300  GO TO 225
305  LET B=B+19
310  LET A=A - B
315  GO TO 180
320  PRINT "NOT ENOUGH FUEL"
325  GO TO 75
```

# A 1K Disassembler

If you have 16K, it will allow you to load Z80 code from various sources and disassemble the code into understandable form. When he was working on the program, Mike soon discovered that 1K would not allow a full disassembler to be written, which is not surprising. Therefore, the program given here does not print mnemonics, but does group the code into separate instructions as follows:

| Address | Code | Mnemonic (not printed) |
|---|---|---|
| 13D | CD5200 | CALL 0052 |
| 140 | 3E FF | LDA,FFH |
| 142 | C9 | RET |

The address of each separate instruction is converted to Hex and the code for one complete instruction is then printed. This renders the code into an easily understood form.

Program Design

The Z80 OP. Codes were grouped into tables, according to whether they were two, three or four byte codes, the remainder being one byte codes. The tables so produced were as follows:

| Table 1 2 byte codes | Table 2 3 byte codes |
|---|---|
| 06 0E 10 16 | 01 11 21 22 |
| 18 1E 20 26 | 2A 31 32 3A |
| 28 2E 30 36 | C2 C3 C4 CA |
| 38 3E C6 CB | CC CD D2 D4 |
| CE D3 D6 DB | DA DC E2 E4 |
| DE E6 EE F6 | EA EC F2 F4 |
| FE | FA FC |

| Table 3 4 byte codes |
|---|
| DD ED FD |



**Mike Biddell has produced a disassembler which just squeezes into 1K on the ZX81. The main aim of the program is to allow you, to unlock the secrets contained in the ROM, so that you can gain a working understanding of the routines indside it.**

These tables were then written into the USR routine (see figure 1.) The op. code byte to be decoded is POKEd into address 16514. The USR routine then examines this byte and compares it with each look up table in turn using the powerful Z80 CPIR instruction. If a true comparison in Table 1 (two byte codes) is found, the BC register pair is loaded with the value 2 (Hex or decimal) and the machine code routine returns to BASIC. With the ZX81, the value of USR is the contents of the BC register pair and therefore if the program RETs at this stage, USR is assigned the value 2. If no match is found in table one, the code jumps to table 2 (three byte codes).

If a match is found, the program Returns with BC, and hence USR, assigned the value 3. Again, if no match with the byte under scrutiny is found, table 3 is examined, where USR is assigned a value of 4. Finally, if no match is found, the byte must be a one byte instruction and USR is unity.

The BASIC program, shown in figure 2, calls the machine code routine and carries out the necessary decimal to hexadecimal conversions. Variable W holds the USR value and prints out 'W' bytes of the code to be disassembled. In this 'disassembled' form the code is quickly and easily understood.

Entering The Program

Type in the machine code loader shown in figure 3. This loader allows code to be POKEd into REM statement. (After REM type in the letter A, 150 times, to reserve space for the code.) Run the program and it will print "16514 38". This indicates the first letter A after the REM. (The code for A is 38.) Now press Newline until address 16520 is reached. The machine code routine is now entered in decimal values (from figure 4). Newline is pressed after each entry. When this has been achieved LIST line 1 to see the REM statement with the

Figure 1.

**USR Routine (Hex)**

| Code | Mnemonic | Comment |
|---|---|---|
| 3A 82 40 | LD A,(NN) | Address of storage byte for code to be disassembled. Put byte into accumulator. |
| 21 00 00 | LD HL,00 | Clear HL |
| 09 | ADD HL,BC | Loads HL with address of start of USR routine. |
| 16 00 | LD D,00 | Clear D |
| 1E 1B | LD E,1B Hex | |
| 19 | ADD HL,DE | Point HL at start of table 1 |
| 01 19 00 | LD BC, 19 Hex | Length of table 1 |
| ED B1 | CPIR | Look for a match in table 1 |
| 28 25 | JRZ | Jump if match found |
| 18 27 | JR | Jump no match |
| 00 00 00 00 | NOP X 4 | Work space |
| 06 0E 10 16 | | |
| 18 1E 20 26 | | |
| 28 2E 30 36 | | |
| 38 3E C6 CB | | |
| CE D3 D6 DB | Table 1 | Two byte op. codes |
| DE E6 EE F6 | | |
| FE | | |
| 00 00 00 00 | NOP X 6 | Work space |
| 00 00 | | |
| 01 02 00 | LD BC,02 | Value of USR |
| C9 | RET | Return to BASIC |
| 01 1A 00 | LD BC,1A | Length of table 2 |
| 1E 1A | LD E,1A | |
| 19 | ADD HL,DE | Point HL at table 2 |
| ED B1 | CPIR | Look for a match |
| 28 24 | JRZ | Jump match found |
| 18 26 | JR | Jump no match |
| 00 00 00 00 | NOP X4 | Work space |
| 01 11 21 22 | | |
| 2A 31 32 3A | | |
| C2 C3 C4 CA | | |
| CC CD D2 D4 | Table 2 | 3 byte op. codes |
| DA DC E2 E4 | | |
| EA EC F2 F4 | | |
| FA FC | | |
| 00 00 00 00 | NOP X 4 | Work space |
| 01 03 00 | LD BC,03 | Value of USR |
| C9 | RET | Return to BASIC |
| 01 03 00 | LD BC,03 | Length of table 3 |
| 1E 18 | LD E,18 | |
| 19 | ADD HL,DE | Point HL at table 3 |
| ED B1 | CPIR | Look for match |
| 28 0D | JRZ | Jump match found |
| 18 0F | JR | Jump no match |
| 00 00 00 00 | NOP X4 | Work space |
| DD ED FD | Table 3 | 4 byte op. codes |
| 00 00 00 00 | NOP X4 | Work space |
| 01 04 00 | LD BC,04 | Value of USR |
| C9 | RET | Return to BASIC |
| 01 01 00 | LD BC,01 | Value of USR (1 byte code) |
| C9 | RET | Return to BASIC |

Figure 2.

**BASIC Program**

```
"DISASS"
  1 REM "MACHINE CODE"
 10 PRINT "DEC.  ADD.?"
 16 LET T = 16520
 20 INPUT V
 22 LET A = V
 23 LET Z = 1
 24 CLS
 25 PRINT "ADD.        CODE"
 26 GOSUB 900
 40 POKE T - 6, PEEK V
 50 LET W = USR T
 60 FOR J = 1 TO W
 70 LET A = PEEK V
 80 GOSUB 900
 90 LET V = V+1
100 NEXT J
110 INPUT A$
120 IF A$ = " " THEN GOTO 22
130 IF A$ = "N" THEN GOTO 20
140 GOTO 110
900 DIM Z(4)
905 LET S = 1
910 LET X = INT(A/16)
920 LET Y = A-16*X
930 LET Z(5-S) = INT(Y+28)
940 LET S = S+1
950 LET A = X
960 IF A>0 THEN GOTO 910
970 FOR I = 1 TO 4
980 PRINT CHR$(Z(I));
981 NEXT I
995 RETURN
```

Figure 3.

**Machine Code Loader**

```
  1 REM (150 letter As)
 10 LET A = 16514
 15 CLS
 20 PRINT A;" ";PEEK A
 25 LET A = A+1
 30 INPUT A$
 40 IF A$ = " " THEN GOTO 15
 50 IF A$ = "N" THEN GOTO 90
 60 IF A$ = "R" THEN GOTO 120
 70 POKE A - 1, VAL AS
 80 GOTO 15
 90 INPUT A
100 GOTO 15
120 CLS
125 FAST
130 LET V = USR(A)
135 LET A = A - 1
140 GOTO 15
```

Figure 4

MACHINE CODE ROUTINE (DECIMAL)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 58 | 13Ø | 64 | 33 | ØØ | ØØ | Ø9 | 22 |
| ØØ | 3Ø | 25 | 25 | Ø1 | 25 | ØØ | 237 |
| 177 | 4Ø | 31 | 24 | 39 | ØØ | ØØ | ØØ |
| ØØ | Ø6 | 14 | 16 | 22 | 24 | 3Ø | 32 |
| 38 | 4Ø | 46 | 48 | 54 | 56 | 62 | 198 |
| 2Ø3 | 2Ø6 | 211 | 214 | 219 | 222 | 23Ø | 238 |
| 246 | 254 | ØØ | ØØ | ØØ | ØØ | ØØ | ØØ |
| Ø1 | Ø2 | ØØ | 2Ø1 | Ø1 | 26 | ØØ | 3Ø |
| 26 | 25 | 237 | 177 | 4Ø | 36 | 24 | 38 |
| ØØ | ØØ | ØØ | ØØ | Ø1 | 17 | 33 | 34 |
| 42 | 49 | 5Ø | 58 | 194 | 195 | 196 | 2Ø2 |
| 2Ø4 | 2Ø5 | 21Ø | 212 | 218 | 22Ø | 226 | 228 |
| 232 | 236 | 242 | 244 | 25Ø | 252 | ØØ | ØØ |
| ØØ | ØØ | Ø1 | Ø3 | ØØ | 2Ø1 | Ø1 | Ø3 |
| ØØ | 3Ø | 24 | 25 | 237 | 177 | 4Ø | 13 |
| 24 | 15 | ØØ | ØØ | ØØ | ØØ | 221 | 237 |
| 253 | ØØ | ØØ | ØØ | ØØ | Ø1 | Ø4 | ØØ |
| 2Ø1 | Ø1 | Ø1 | ØØ | 2Ø1 | | | |

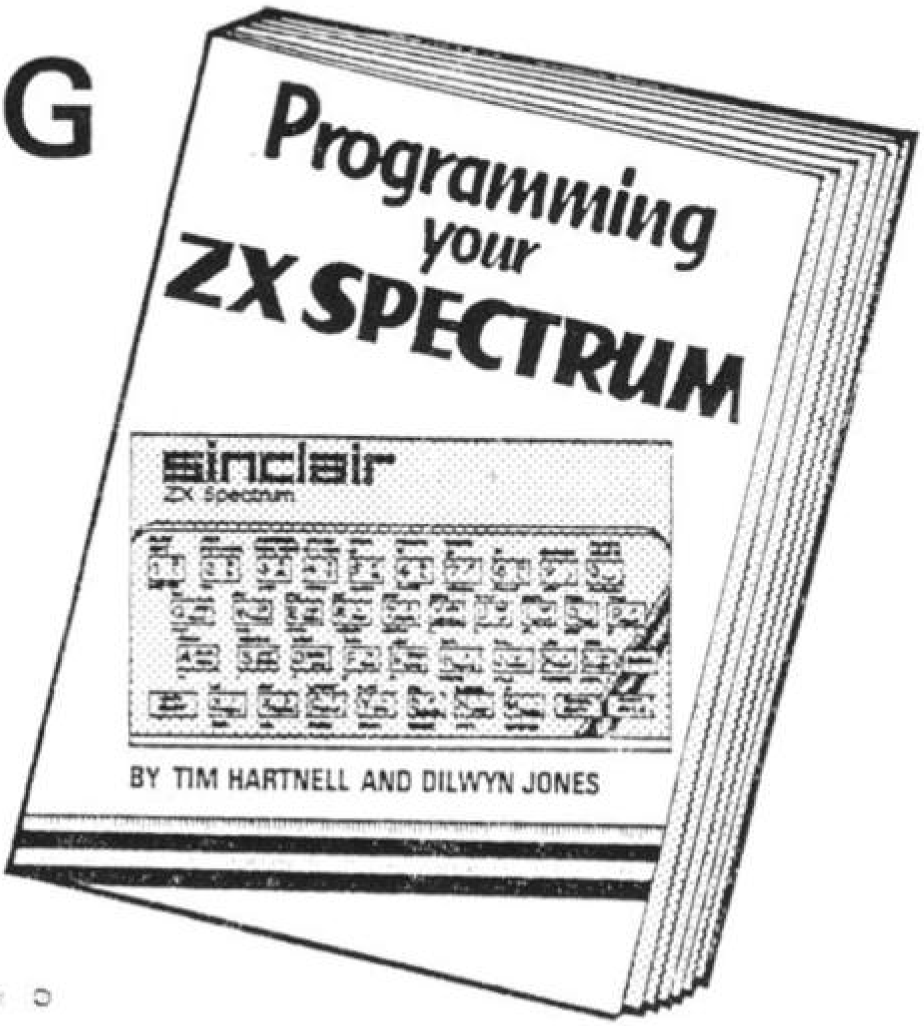
values POKEd into it. (With the ZX81, displaying the REM will not cause a system crash.) If you make an error entering the code, simply press "N" (for new address) Newline, then enter the address at which the mistake was made (followed by Newline). The correct value can then be typed in.

The machine code loader is now deleted (leave the REM statement) in the normal way. The BASIC program shown in figure 2 is now entered. The operation is very simple; when run, the program requests the decimal address in ROM or RAM at which disassembly should start. To test the program, start at 16520 and with each press of Newline, the program will disassemble itself as shown in figure 1. (Except when it reaches the tables.) The program disassembles data tables as though they were program and there is some ambiguity surrounding the four byte codes. However, for the most part, the program works extremely well, rendering meaningless code into understandable form.

The disassembler tends to be self aligning, ie if you jump into code at a data byte, rather than an op. byte, it tends to sort out the programming logic after about three presses of Newline and 'tune in' to the op. bytes.

The disassembler has unravelled vast chunks of the ZX81 ROM for the author and should be an invaluable programming aid for machine code writers using only the 1K machine and for whom, available 4K or so, disassemblers are not a practical proposition.

# THE BOOK YOU'VE BEEN WAITING FOR!




Examines and explains every function on the ZX Spectrum

Just £6.95

Try this program from the book:

```
 5 PAPER 0: CLS : BORDER 0
10 FOR X=0 TO 255
20 PLOT X,0
30 DRAW OVER 1,255-X*2,175
40 NEXT X
50 FOR Y=0 TO 175
60 PLOT 0,Y
70 DRAW OVER 1,255,175-Y*2
80 NEXT Y: REM © C. Ruston
```

Takes you through programming your ZX Spectrum from first principles right through to such things as defining your own graphics, MERGE, READ/DATA, SCREEN$, POINT and DRAW.

Interface,
Dept ZC
44–46 Earls Court Road
London W8 6EJ.

Please send me the following:

[ ] Programming Your ZX Spectrum – £6.95
[ ] A sample issue of INTERFACE, the monthly ZX magazine published by the National ZX Users' Club – £1.00
[ ] Getting Acquainted with your ZX81 – £5.95
[ ] 20 Simple Electronic Projects for the ZX81 – £6.45
[ ] Mastering Machine Code on your ZX81 – £7.50
[ ] 34 Amazing Games for the 1K ZX81 – £4.95
[ ] 49 Explosive Games for the 1K ZX81 – £5.95
[ ] The Gateway Guide to the ZX81 and ZX80 – £6.45

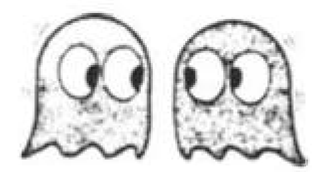
I enclose a total of £ ____________

Name ______________________________

Address ____________________________

____________________________________

THIS BOOK IS TOTALLY ORIGINAL. IT IS NOT BASED ON ANY OTHER BOOK, & THE PROGRAMS ARE ALL NEW



User-defined graphics chapter tells you how to create your own DOTMAN game!

# Caring for your computer

**Reviewer Alex Heywood takes a selection of books from ZX and computer library shelves and assesses their value for owners of Sinclair computers.**

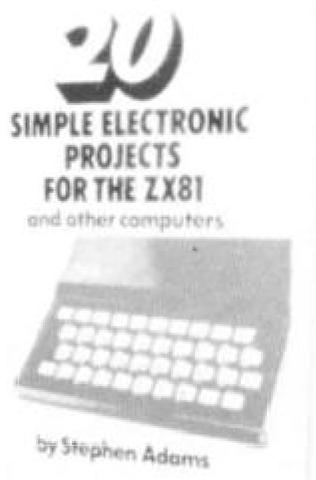
## DON'T (or How to Care for your Computer)

Written by Dr. Rodnay Zaks, the leading light of Sybex, who've published the book, "DON'T" is designed to tell you how to care for your computer, and how to avoid doing things which could damage it. In America many computer manufacturers have been buying the book in bulk to supply with their computers to customers. They have discovered that the cost of the book is minimal compared to the savings in service calls it produces.

The book is aimed predominently at buyers of business systems although there is specific information aimed squarely at home computer owners. Each chapter starts with a section headed for the home computer user, and although some of the advice seems like common sense it can well bear repeating.

In a section headed 'The Computer Room', which tells commercial operators of optimum operating conditions for their computers, Dr. Zaks says to the home user: "Keep the room comfortable for a human. Your computer will like it, too." A simple statement, but one which bears thinking about as it may well suggest further thinking. If a room is too cold for a human to work in, what is it doing to the computer?




The advice on discs does not apply (yet) to ZX owners, though with the advent of Clive's Microdrive it may well do so. There are four specific bits of advice given on working with floppy discs: Protect each new diskette; insert the disc correctly; follow the proper power-up/power-down procedure; and inspect discs each time they are used. We may well find that advice invaluable when the Microdrive becomes available.

Overall this book is not aimed at a ZX owner, but it makes interesting reading, and could well make you an expert at advising others on what potential sins they are committing.

*DON'T (or How to Care for your Computer)*, Dr. Rodnay Zaks, Sybex, ISBN 0-89588-065-2.

## 20 Simple Electronic Projects for the ZX81

This book, published by INTERFACE, who are well entwined with the National Users' Group, is aimed squarely at the ZX owner, but also caters for those who have computers other than the ZX81. Author Stephen Adams, well known for his construction articles in computer magazines, and for his reviews of ZX add-ons, lists 85 computers (as well as the ZX81) which can use the pro jects. Well illustrated with circuit diagrams and photographs (plus a few rogue photos of Atoms, BBC Micros and MZ-80Ks) the book assumes no previous knowledge on the part of the person who will construct the projects. Advice on such apparently mundane matters as the correct way to solder, and how to read resistor colour codes, points out that everybody has to start at the beginning.

The projects are varied, and to my inexperienced eye, seemed to represent a range of projects, from the very simple to the more complex. To give you an idea of the kind of book it is, I shall list the projects: Mains operated 5 volt/12 volt power supply; a monitor; a universal gate; tape recorder control; minitone; numeric keypad for the ZX81; giant seven segment display; score board; wheel of fortune; analogue to digital converter (A/D); light pen; shift lock for keyboards; a cheap thermometer (if you ignore the cost of the computer!); graphics — function — edit — rubout key for the ZX81; the movable 'occupant'; "unbeatable" burglar alarm; standby power supply; mains supply filter; a logic probe. The contents also include a number of diagrams of basic components (although I imagine most of us already have a pretty good idea of what a loudspeaker looks like), resistor and capacitor colour codes, and useful addresses.

If you are at all interested in building peripherals for your ZX81, this is obviously the place to start. The text and circuit diagrams are clear; the photographs give you some idea of what the project will look like when completed; and no prior knowledge is assumed.

*20 Simple Electronic Projects for the ZX81 and other computers* — Stephen Adams, Interface, ISBN 0 907563 11 2.

## Fifty BASIC Exercises

Published by Sybex, this 226-page book by J. P. Lamoitier, whose field of expertise is the use of FORTRAN and BASIC in business environments, is designed for those who know BASIC, but would either like to improve their programming or get additional ideas for programs — or both.

Starting with the inarguable statement 'the best way to learn a computer language is through actual practice', Mr. Lamoitier takes the reader through a series of completely explained exercises: statement and analysis of the problem, flowcharts, programs and actual runs. This format will help you improve your programming.

The programs are carefully and exactly coded. There is no "it runs so we'll print it" thinking which appears evident in some other books. Examining the listings and reading the text

will teach you quite a bit about how to improve your programs, even if you do not bother to adapt all of them for the ZX81 or Spectrum. Programs in the fields of mathematics, business, operations research and games, presented in varying levels of difficulty, have been chosen for their 'educational' value as much as for their relevance to everyday applications.

Many of us who quite enjoy programming, and have developed a fair degree of facility at doing so, need ideas to spark off new programs. This book is a great source of such ideas. Programs and other items in the book include: The purpose of a flowchart and how to verify one; a 'flip-flop' technique for branching; Armstrong numbers; conversion from base ten to another base; determination of a circle passing through three given points; plotting a curve; calculation of a definite integral; numerical evaluation of polynomials; sales forecasting; 'Matchstick Game'; Craps; topological sort; linear regression; and the Eight Queens Problem.

Overall this is a carefully written book which, if studied and the programs converted to run on your computer, must enhance your programming ability.

*Fifty BASIC Exercises,* J. P. Lamoitier, Sybex, 0-89588-056-3.

## First Book of Party Tricks for the ZX81 (1K)

Published by Video Software Ltd., this slim (26 pages) volume should not be sneered at because of its unimpressive size or presentation. The programs are, on the whole, original in concept and implementation, and are documented in great detail, which helps a lot in trying to work out what part of a program does what.

The programs are: Shoot; Sketch; Name the Day; Train; Onger-Wonger; Weather; UFO; Who Shot JR; Field Gun; and Follow Hat. Ignoring the less original ones, with titles such as 'Sketch', I decided to have a look at the ones which showed a truly creative approach to the problem of squeezing a program into 1K.

In SHOOT, you are about to take a penalty, and the goalkeeper is waiting for you. Press any key to shoot and the goalkeeping attempts to save your shot. A running total of your goals and his saves is kept by the ZX81. The most interesting thing about this program (apart from the fact that you'd need to be told what each symbol represents, which is neither surprising nor important) is a line which saves a considerable amount of space by using the method the ZX81 works out logical expressions. The line, 350, reads: LET J = A + (S A OR S = T OR S (B + B)), which changes the value of the A which is assigned to J only if one or more of the conditions within the brackets are true.

TRAIN produces a little train which obeys the "5" (move left) and "8" (move right) keys. Fun to watch, but that's about all. The program which follows it in the book, Onger-Wonger, shows how to make the computer draw its own pictures. In this case, it draws an Onger-Wonger bird which flies around the screen in ever-decreasing circles. Once you've run that, you can use the program for storing your own pictures as a line of 'data' as elements in a string. The computer accesses the string element by element, and as a result of what it finds in the string, draws the picture. This program would usefully serve as a subroutine in a much larger program to set up the starting scene.

*First Book of Party Tricks for the ZX81 (1K),* Philip Smith, Video Software, no ISBN.

## Personal Computers Handbook

This book is aimed at those who are afraid of computers! At least author Walter Buchsbaum says so in his introduction, and adds that if you are in this situation you are not alone. ". . . most people in your circle of acquaintances are (also) likely to be, at least, a little afraid of computers," he claims. While this introduction seems to have little relevance to the balance of the book, it does give Mr Buchsbaum a chance to explode some myths and misconceptions about computers, including discussing the exact meaning of the word 'intelligence' when applied to machines, and whether a computer can be 'smarter' than a person or display 'talent'.

This matter firmly dealt with, the book goes on to describe 'what computers really do', leading through a discussion of binary numbers to how microprocessors work, and the architecture of some common memory chips.

As can be seen from the brief discussion so far, this book is somewhat different from the others reviewed in this section of the magazine. It assumes the reader has some fundamental questions about computers which need answering, than a need for an understanding of how modern microcomputers are constructed. The book is clearly aimed at a person who does not yet own a personal computer but is thinking of buying one, and reviews a number of small systems, including the TRS-80 and the Atari 400, to show which applications each machine best supports.

You'll find it an interesting book, not least for the programs (such things as an arithmetic quiz program, and a bubble sort) but for the overview of the personal computer market the book provides.

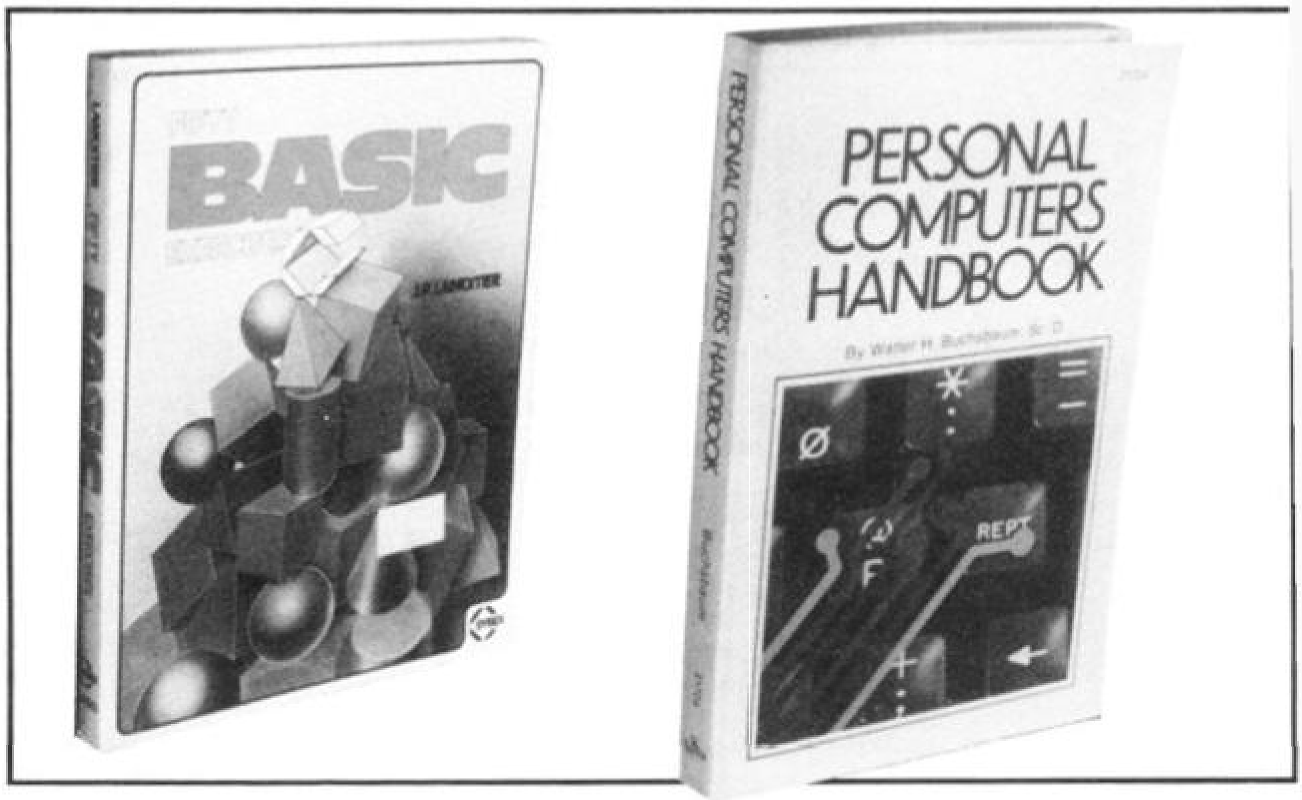
*Personal Computers Handbook,* Walter Buchsbaum, Howard W Sams and Co., ISBN 0-672-21724-4.

## The Explorers Guide to the ZX81

Written by Mike Lord, guiding light of Timedata, this book is firmly in the tradition of *The ZX80 Magic Book,* and *The Atom Magic Book*. . . only it is much thicker than its predecessors.

There can be no argument about the value of the contents. From 'Converting other BASICs' to 'Building your own 16K RAM', there is much to interest and instruct the ZX81 owner. The only problem — if it is worth calling a problem — is that there appears to be little coherent link between the six sections of the book. One suspects Mr Lord simply got everything he knew about the ZX81, divided into six separate piles, and then called it a book.

But no matter. The lack of structure to the volume is relatively unimportant, although it



is a characteristic of earlier Timedata books. The contents are the most important aspects, regardless of how they are arranged. Several programs (including the worthwhile simulation exercise in which you have to run a computer software store) are worth the cost of the book alone, while the hardware section is sure to appeal to soldering iron buffs. Adding an additional keyboard, connecting a monitor and overcoming spiky mains are among the items discussed in an expert manner in this book.

The section entitled DISCOVERING THE ROM includes a run down of many of the important ROM routines which can be called by machine code programmers. The LOAD and SAVE routines are discussed and the action of the display file is outlined. The major circuit elements involved in producing the display are investigated and this leads well into the description of how to improve the picture by connecting up a monitor.

All in all, this book will prove a worthwhile resource for the ZX81 owner, with the ROM and hardware information of the greatest value.

*The Explorers Guide to the ZX81*, Mike Lord, Timedata, no ISBN.

## Out of the mouths of 10-year-olds

Ten-year-old A. D. Lindsay of Frodsham sent us in his review of a number of ZX books, and we include the review here, word for word as we received it.

A couple of days ago a few packages dropped through my letterbox. They did (surprise, surprise) all contain books.

It is amazing what books can do. They re-awaken interest in the black-shelled object, which just happens to be called ZX81. There are lots of ZX81s lying neglected in their boxes because, frankly, there is nothing else to be done with them.

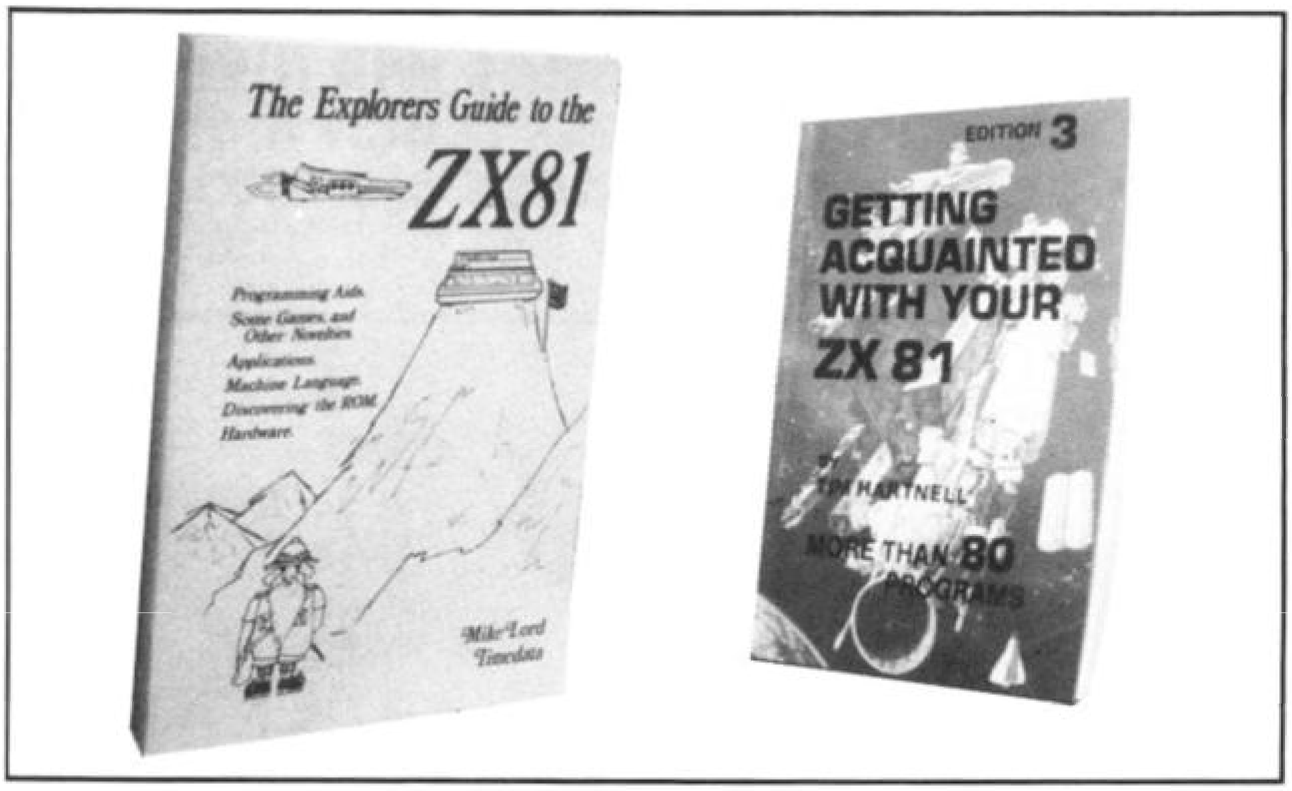
You do, however, have to be careful with books; some of the programs do not work. All of the ones that I tried worked first time — or were quite easy to modify.

The first book which came under my scrutiny was called *49 Explosive Games for a ZX81*. This book does not (as you will have gathered from the title) pretend to teach you programming; it just has page after page of great programs. Some are real oldies, like 'NIM' but some are completely new, like an adventure called 'Smugglers Bold' which fits in to 8K RAM '81.

The programs are chosen, I am told (it says on the back) to illustrate a certain programming skill. There is, however, a conversion table of PEEKs and POKEs from the old ZX80 ROM to the new 8K ROM. This is very useful when converting 27 ZX80 programs to run on the 81. Many of the programs need extra memory so I would advise people with only 1K RAM to leave this book alone.

The next one I examined was called *Not Only 30 Programs for the Sinclair ZX81*. This book is a collection of programs which show just how much can be crammed into Sinclair's tiny RAM chips. Most of the programs are games. The first one is a very basic pattern generator, though a 1K Draughts is available later in the book. The programmer, however, has to report to machine code for this game to fit into Clive's magic chips.

They are written by a few different programmers but are all set out in the same way. First there is a description of what happens when the program is run. After this, the structure — or how the program is built — is explained. Then there are notes on running it and finally, the actual program.




The *ZX81 Pocket Book* is much better than its ZX80 counterpart. Trevor Toms has learnt from his mistakes and the book now has more space-saving hints and games. One very useful routine which he discovered was PAUSE 4E4. This means that the computer pauses until a key is pressed.

*Getting Acquainted with your ZX81* is written by Tim Hartnell. This is probably the best ZX81 book around because it combines games with tips. It contains over 80 programs. Many of these are very short, but there are a few good games. If "PRINT PEEK 16396 + 256 * PEEK 16397-16509" is used as a direct command, it will print how many bytes of memory have been used in your program. This can be very useful if you have only 1K RAM to play with. There are a few graphics routines which show just what can be obtained with a six-line program. These are listed under titles such as Games and Data Files. For someone with only 1K it is a good investment. The book shows just how much can be squeezed into the 81's mini memory. If one has 16K, one would be critical of some of the games, though one could modify to make them even better.

*Understanding Your ZX81 ROM* aims to teach you how to program short machine code routines into your computer. It contains 26 basic programs and quite a few chapters. Chapter Six examines the 8K monitor in quite some detail. This is written by Dr. Logan and it is really for people with no previous machine code knowledge. Those who have will be very disappointed not to find a complete listing of the ROM in one place.

The *ZX81 Companion* is written for people with the 16K RAM pack. Without it, this book is of no use. It is not a "games" book but rather a learning book. Some of the chapters are about saving and loading machine code and the ZX81 as an educational tool.

Chapter Four deserves a special mention because it is very good — "Examining and using the Monitor" is its title and it contains an almost complete monitor listing.

(I shall give each book a star rating, out of five):

49 EXPLOSIVE GAMES FOR A ZX81 * * * *
Written by Tim Hartnell, published by Interface. £5.25.

NOT ONLY 30 PROGRAMS FOR THE SINCLAIR ZX81 * * *
Written by various authors and published by Melbourne House. £6.95.

THE ZX81 POCKET BOOK * * * *
Written by Trevor Toms and published by Phipps Associates £4.94.

GETTING ACQUAINTED WITH YOUR ZX81 * * * * *
Written by Tim Hartnell, published by Interface. £5.95.

UNDERSTANDING YOUR ZX81 ROM * * *
Written by Dr. Logan, published by Melbourne House. £8.95.

THE ZX81 COMPANION * * * *
By Bob Maunder, published by Linsac. £7.95.

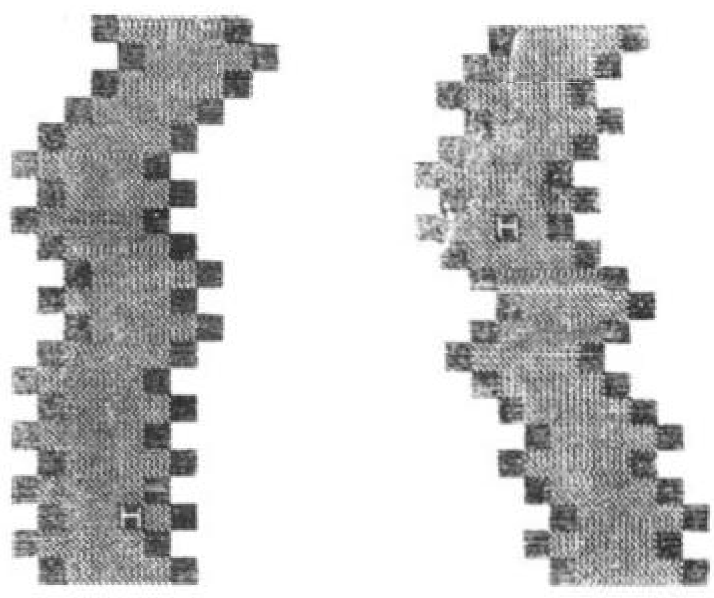
# Twisting and turning

**One of the most common complaints about the ZX81 concerns the 1K provided with the standard computer. Skilful programming can get around this apparent lack of memory as these programs show.**
**You'll find that studying the listings will give you ideas on how you can compress much more program than you thought possible into the 1K ZX81.**

## Alley Driver

In Alley Driver, written by Said Hasson of Worthing, you have to drive a car down a constantly twisting track. Said explains: "The idea for the pro gram is not really original, I know, but I think the way I've done it in this game is. Instead of scrolling the screen to give a racing car effect, as you explained in your article on Moving Graphics in the last issue of ZX Computing, the car (an inverse 'H') races down the screen. The effect, I feel, is slightly smoother and faster than using 'scroll'.

"After each section is completed, the screen clears and a new track appears. The program supports a high score feature, and after each game will ask the player if he or she wishes to have another game. Pressing "Y" will produce a new game."



```
  10 LET H=CODE ""
  20 LET S=CODE ""
  30 CLS
  40 LET X=CODE "■"
  50 LET A=CODE "■"
  60 FOR N=CODE "" TO CODE "="
  70 PRINT TAB A;"■▒▒▒▒▒▒■"
  80 LET A=A+(A<CODE "■" AND A)*
SGN (RND-.5)+(NOT A)-(A=CODE "■"
)
  90 NEXT N
 100 LET A=X
 110 FOR N=PI/PI TO CODE "="
 120 PRINT AT N,X;
 130 IF PEEK (PEEK 16398+256*PEE
K 16399)=CODE "■" THEN GOTO 210
 140 PRINT AT N-PI/PI,A;"▒";AT N
,X;"H"
 150 LET A=X
 160 LET X=X+(INKEY$="0")-(INKEY
$="1")
 170 NEXT N
 180 CLS
 190 LET S=S+N
 200 GOTO CODE "2"
 210 CLS
 220 LET S=S+N
 230 PRINT ,,,"[illegible]",,,"SCORE
=";S,
 240 IF H<S THEN LET H=S
 250 PRINT "HIGH SCORE=";H,,,"PL
AY AGAIN?"
 260 IF INKEY$="" THEN GOTO 260
 270 IF INKEY$="Y" THEN GOTO COD
E "="
```

## Sorting it out

This utility program, from Ann Marshall of Coventry, sorts a series of numbers (positive, negative or mixed) into order. When you run the program, you'll get a prompt as the computer waits for you to enter the number of items you want sorted. Enter this number, then press NEWLINE/RETURN, and then enter the items of data one by one. Once they are all in, the computer will sort them, then print them in order, numbering each one as it prints them out. As it is now, the computer sorts the numbers into descending order. If you want them in ascending order, then reverse the 'greater than or equal to' sign in line 130.

```
  5 LET Y=PI/PI
  6 LET X=VAL "5"
 10 LET A=X
 30 INPUT N
 40 DIM A(N+X)
 50 FOR T=A TO N+X-Y
 60 INPUT A(A)
 90 LET A=A+Y
100 NEXT T
110 FOR B=X TO A-Y
120 FOR C=B+Y TO A-Y
130 IF A(B)>=A(C) THEN GOTO 170
140 LET D=A(B)
150 LET A(B)=A(C)
160 LET A(C)=D
170 NEXT C
180 NEXT B
190 FOR B=X TO A-Y
195 SCROLL
200 PRINT B-X+Y,A(B)
210 NEXT B
```

## Permutating

Our third program, another one by Said Hassan, calculates combinations and permutations. You are first asked which calculation you want to perform.

Lines 300 to 400 check that the input figures are numerical and lie within the machine's capabilities. The permutation of taking n different items r at a time is given by the formula:

$$nPr = \frac{n!}{(n-r)!}$$

(n! is n factorial)

For example, consider five different playing cards that have to be arranged in groups of three

n = 5, r = 3, and 5P3 = 60.

The combination of taking n items r at a time is given by the formula

$$nCr = \frac{n!}{(n-r)!\,r!}$$

How many ways can three book titles be selected from five book titles?

5C3 = 10.

```
 10 PRINT """C"" COMBINATIONS",
"""P"" PERMUTATIONS"
 20 LET A$=INKEY$
 30 IF A$="" OR (A$<>"P" AND A$
<>"C") THEN GOTO 20
 40 PRINT "INPUT N"
 50 GOSUB 300
 60 LET N=VAL N$
 70 PRINT " INPUT R"
 80 GOSUB 300
 90 LET R=VAL N$
100 IF R<=N THEN GOTO 130
110 PRINT "TOO HIGH"
120 GOTO 70
130 PRINT "CALCULATING"
140 LET Z=N
150 GOSUB 410
160 LET B=A
170 LET Z=N-R
180 GOSUB 410
190 LET C=A
200 IF A$="P" THEN GOTO 240
210 LET Z=R
220 GOSUB 410
230 LET C=C*A
240 CLS
250 PRINT N;" ";A$;" ";R;" = ";
B/C
260 IF INKEY$="" THEN GOTO 260
270 CLS
280 RUN
300 INPUT N$
310 IF N$="" THEN GOTO 300
320 CLS
330 FOR X=1 TO LEN N$
340 IF (N$(X)>="0" AND N$(X)<="
9") THEN GOTO 370
350 PRINT "ERROR RE-ENTER"
360 GOTO 300
370 NEXT X
380 IF VAL N$<34 THEN RETURN
390 PRINT "TOO HIGH, RE-ENTER"
400 GOTO 300
410 LET A=1
420 FOR X=1 TO Z
430 LET A=A*X
440 NEXT X
450 RETURN
```

## Getting primed

Our final 1K program in this section is a way of getting your ZX81 to earn its living generating prime numbers. When you run the program you'll get a prompt. This is the number of prime numbers you want the computer to generate for you. It will then proceed to do so for you, printing them out as it works them out. If you want a permanent record of your computer in its prime, change line 160 to read LLPRINT D.

We modified the program slightly to count the number of primes it had generated, and after running it over *five hours* had only got to prime number 6030 (see printout). The ZX81 was getting pretty hot by then so we stopped the process. A pity, as we'd love to find out what the 10,000th prime is.

```
  5 LET X=PI/PI
  7 LET Y=X+X
 10 LET D=Y+X
 15 SCROLL
 20 PRINT "HOW MANY
 30 INPUT A
 35 SCROLL
 40 PRINT A,1;" ";2;
 60 FOR B=X TO A-D
 70 LET D=D+Y
 80 LET C=Y+X
 90 LET E=INT (D/C)
100 LET F=D-E*C
110 IF F=0 THEN GOTO
120 IF C>=E THEN GOT
130 LET C=C+Y
140 GOTO 90
150 SCROLL
160 PRINT ,D
170 NEXT B
```

```
6030        59651
6031        59659
6032        59663
6033        59669
6034        59671
6035        59693
6036        59699
6037        59707
6038        59723
6039        59729
6040        59743
6041        59747
6042        59753
6043        59771
6044        59779
6045        59791
6046        59797
6047        59809
6048        59833
6049        59863
6050        59879
6051        59887
```

# Lining up numbers

## There is something irritating about a list of numbers display in a tatty and irregular format. Nick Godwin from Eyemouth, Berwicks decided to do something about it.

Consider the following versions of the same sum:

| | |
|---|---|
| 99.089 | 99.09 |
| 679.0734 | 679.07 |
| −2 | −2.00 |
| 679 | 679.00 |
| −186 | −186.00 |
| 46.009 | 46.01 |
| −269.087 | −269.09 |
| −12 | −12.00 |
| 148 | 148.00 |
| 981.08 | 981.08 |
| 2163.1644 | 2163.1644 |

The version on the left was produced by the following routine:

```
100  LET B=0
110  FOR J=1 TO 10
120  INPUT A
130  LET B=B+A
140  PRINT A
160  NEXT J
170  PRINT
190  PRINT B
199  STOP
```

into which, of course, I entered the values which I wanted to be summated.

This is very untidy.

To start dealing with the problem, modify the program by adding or changing certain lines, as follows:

```
105  LET T=16
140  LET X=A
150  GOSUB 1000
180  LET X=B
190  GOSUB 1000
```

The value of T can be adjusted to change the lateral print position, but be sure to allow sufficient room on the left of the screen for the longest number you want to enter.

The following subroutine, applied to the above, is suitable if you only wish to enter positive integers:

```
1010 LET X$=STR$ X
1020 PRINT TAB T-LEN X$;X$
1029 RETURN
```

If you wish to enter decimal numbers, but only want its nearest integer printed in each case, add the following lines:

```
1000 IF X=.5 THEN LET X=.6
1005 LET X=INT (X+.5)
```

You may want to be able to enter either integer or non-integer values, and to have these printed in full, in which case substitute the following for the whole of subroutine 1000 above.

```
1010 LET X$=STR$ X
1020 IF X$(1)="." THEN LET
     X$="0"+X$
1030 FOR K=1 TO LEN X$
1040 IF X$(K)="." THEN
     GOTO 1070
1050 NEXT K
1060 LET X$=X$+".0"
1080 PRINT TAB T-K;X$
1089 RETURN
```

You may wish to be able to enter negative values, in which case add the following line:

```
1015 IF X (less than) 0 THEN IF
     X$(2)="." THEN LET
     X$=X$(1)+"0"+X$
     (2 TO)
```

You may wish to print only the first n decimal places. For example, the addition of the following lines would be suitable for cash (ie. two decimal places):

```
1002 IF X=.005 THEN LET
     X=.006
1003 IF X=-.005 THEN LET
     X=-.006
1005 LET X=INT (100-X+.5)
     /100
1070 IF X$(LEN X$-1)="."
     THEN LET X$=X$+"0"
```

You may wish to put, as I have done in my opening example, the total in complete form, in which case add the following line:

```
1006 IF J=11 THEN LET X=B
```

Another improvement to presentation consists of the addition of the following line:

```
1000 SCROLL
```

You must also amend any PRINT in the main body of the program to SCROLL (ie. line 170).

Finally, here is a progran based on the preceding seg ments of program which ac cepts five numbers, adds then together, and — as can be seer from the print out below the listing — prints them out attrac tively.

```
 100 LET B=0
 105 LET T=16
 110 FOR J=1 TO 5
 120 INPUT A
 130 LET B=B+A
 140 LET X=A
 150 GOSUB 1000
 160 NEXT J
 170 PRINT
 180 LET X=B
 190 GOSUB 1000
 199 STOP
1010 LET X$=STR$ X
1020 IF X$(1)="." AND VAL (X$)>.
0999999 THEN LET X$="0"+X$
1030 FOR K=1 TO LEN X$
1040 IF X$(K)="." THEN GOTO 1070
1050 NEXT K
1060 LET X$=X$+"."
1070 IF X$(LEN X$-1)="." THEN LE
T X$=X$+"0"
1080 PRINT TAB T-K;X$
1089 RETURN

                  123.654
                     .009
                    2.05
                   12.60
                23333.80

                23472.123
```

## Spy Time

If you ever decide to take part-time employment as a spy, your ZX81 could help you get messages to and from enemy territory. The following program for the ZX81 has been adapted from a ZX80 program in the book *Stretching Your ZX81 or ZX80 to its Limit* (published by Computer Publi cations). After the listing ar two sample messages; the firs produced by entering a key number of 193, the second by entering 192.

```
   5 REM ENCODE/DECODE
  10 DIM B(32)
  20 LET D=0
  30 PRINT "ENTER KEY NUMBER(0 T
O 193)"
  40 INPUT A
  45 IF A<0 OR A>193 THEN GOTO 4
0
  50 PRINT "ENTER MESSAGE TO BE
CODED (UP","TO 32 LETTERS)"
  60 INPUT A$
  70 CLS
  80 FOR C=1 TO LEN A$
  90 LET D=D+1
 100 LET B(C)=CODE A$+A
 110 PRINT CHR$ B(C);
 120 LET A$=A$(2 TO )
 130 NEXT C
 140 PRINT
 150 PRINT
 160 INPUT A$
 170 FOR C=1 TO D
 180 PRINT CHR$ (B(C)-A);
 190 NEXT C

 IF INPUT LOAD RAND AT LOAD RAND
 AT SCROLL AT IF FOR RAND IF AT
IF PRINT AT IF SAVE COPY AT LOAD
 IF AT PRINT CLS IF

THIS IS A TEST TO TRY IT OUT
```

# Colourthello

## Challenge your Spectrum to a game of Reversi with this program Colourthello, written by Graham Charlton.

Colourthello is intended to highlight the sound and colour potential of the ZX Spectrum. You'll see when you run the program how effective the new features on the Spectrum can be. You move by entering the number down the side, followed by the number across the top or bottom, as a single two-digit number. For example, if you wanted to place a piece where the bottom "O" is on the board, you'd enter 64.

```
   1 REM ____Colourthello____
   5 PRINT AT 0,12; INK 2;"C"; I
NK 1;"o"; INK 6;"l"; INK 3;"o";
INK 4;"u"; INK 5;"r"; INK 2;"t";
 INK 1;"h"; INK 6;"e"; INK 3;"l"
; INK 4;"l"; INK 2;"o"
  10 DIM a(10,10): FOR b=1 TO 10
: FOR c=1 TO 10
  20 BEEP .01,b*c/10
  40 IF b<>1 AND c<>1 AND b<>10
AND c<>10 THEN LET a(b,c)=CODE "
."
  50 NEXT c: NEXT b: LET p=0: LE
T r=0
  70 LET a(5,5)=CODE "x": LET a(
6,6)=CODE "x": LET a(6,5)=CODE "
o": LET a(5,6)=CODE "o"
 120 INPUT ( INK 2;"Do you want
to go first?  ");q$
 125 CLS : GO SUB 3000
 127 PRINT AT 0,12; INK 2;"C"; I
NK 1;"o"; INK 6;"l"; INK 3;"o";
INK 4;"u"; INK 5;"r"; INK 2;"t";
 INK 1;"h"; INK 6;"e"; INK 3;"l"
; INK 4;"l"; INK 2;"o"
 130 IF CODE q$<>CODE "n" AND CO
DE q$<>CODE "N" THEN GO TO 2000
1000 PRINT INK 2;AT 10,16;"My go
"
1010 LET s=CODE "o": LET t=CODE
"x": LET h=0
1040 FOR a=2 TO 9: FOR b=2 TO 9
1060 IF a(a,b)<>CODE "." THEN GO
 TO 1320
1070 LET q=0: FOR c=-1 TO 1: FOR
 d=-1 TO 1: LET k=0: LET f=a: LE
T g=b
1130 IF a(f+c,g+d)<>s THEN GO TO
 1180
1140 LET k=k+1: LET f=f+c: LET g
=g+d: GO TO 1130
1180 IF a(f+c,g+d)<>t THEN GO TO
 1200
1190 LET q=q+k
1200 NEXT d
1210 NEXT c
1220 IF f=2 OR f=9 OR g=2 OR g=9
 THEN LET q=q*2
1230 IF f=3 OR f=8 OR g=3 OR g=8
 THEN LET q=q/2
1260 IF (f=2 OR f=9) AND (g=3 OR
 g=8) OR (f=3 OR f=8) AND (g=2 O
R g=9) THEN LET q=q/2
1280 IF q<h OR q=0 OR (RND>.3 AN
D q=h) THEN GO TO 1320
1290 LET h=q: LET m=a: LET n=b
1320 NEXT b
1330 NEXT a
1340 IF h=0 AND r=0 THEN GO TO 5
000
1350 IF h=0 THEN GO TO 1370
1360 GO SUB 4000: GO SUB 3000
2000 PRINT INK 1;AT 10,16;"Your
go"
2010 LET s=CODE "x": LET t=CODE
"o": INPUT r
2040 IF r=0 THEN GO TO 2090
2050 IF r<11 OR r>88 THEN GO TO
2030
2060 LET m=INT (r/10)+1: LET n=r
-10*INT (r/10)+1
2080 GO SUB 4000
2090 GO SUB 3000: GO TO 1000
3000 PRINT AT 5,0;: BEEP .25,RND
*5
3010 LET c=0: LET h=0
3030 PRINT INK 4;" 12345678 "
3040 FOR b=2 TO 9: PRINT INK 4;b
-1;
3060 FOR d=2 TO 9
3070 IF a(b,d)=CODE "x" THEN PRI
NT INK 2;"x";
3075 IF a(b,d)=CODE "o" THEN PRI
NT INK 1;"o";
3077 IF a(b,d)=CODE "." THEN PRI
NT INK 5;".";
3080 IF a(b,d)=CODE "x" THEN LET
 c=c+1
3090 IF a(b,d)=CODE "o" THEN LET
 h=h+1
3100 NEXT d
3110 PRINT INK 4;b-1
3120 NEXT b
3130 PRINT INK 4;" 12345678 "
3150 PRINT '' INK 3;"I have "; I
NK 2;c; INK 3;"   You have "; IN
K 1;h;"   "
3170 RETURN
4000 FOR c=-1 TO 1
4010 FOR d=-1 TO 1
4020 LET f=m: LET g=n
4040 IF a(f+c,g+d)<>s THEN GO TO
 4080
4050 LET f=f+c: LET g=g+d: GO TO
 4040
4080 IF a(f+c,g+d)<>t THEN GO TO
 4140
4090 LET a(f,g)=t: IF m=f AND n=
g THEN GO TO 4140
4110 LET f=f-c: LET g=g-d: GO TO
 4090
4140 NEXT d: NEXT c: RETURN
5000 IF c>h THEN PRINT "I won, "
;c;" - ";h
5010 IF h>c THEN PRINT "You won,
 ";h;" - ";c
5030 © Charlton 1982
```

```
 12345678
1....X...1
2...X....2
3..XOX...3
4...OX...4
5..XOX...5          Your go
6...O....6
7........7
8........8
 12345678

I have 7   You have 4
```

# EZUG rides on

**Eric Deeson has been running EZUG, (the Educational ZX User Group) for two years now, since not long after the ZX80 invaded our space. He reckons that the Group is the world's largest for teaching with a specific micro; the number of folk on the list is now about 1500 and almost 10% of them live outside Britain.**

Tim Hartnell, who once described EZUG as sounding like a detergent, asked Eric how well EZUG has met its original objectives.

"We had only one aim at the beginning," was the reply. "It was to provide the MUSE Software Library with good ZX80-based teaching programs."

MUSE is, you ought to know, a large active British association for educational computing. Its magazine, *Computers in Schools* (published by Heinemann) is very readable, but the members seem to find the Software Library a major benefit.

At the time EZUG was started, the Software Library contained material mainly for the PET, the Research Machines 380Z and the TRS 80. The Library has grown steadily, but at the time we spoke to Eric he proudly noted that Sinclair material was pushing hard for the number one spot. There were then forty 16K ZX81 programs in the list with as many more going through the rather arduous assessment procedures.

"No, I can't really cope with the work," said Eric, pointing at a cardboard box full of unlooked-at cassettes.

"At the moment I have to spend at least an evening a week just duplicating the cassettes ordered by members."

That task is unpaid, so the obvious solution (handing it over to the pupils at school) is not feasible. "I can't expect my students to do a dreary job like that for nothing."

"You haven't mentioned the Newsletter," pointed out Tim.

The Newsletter started off in concept simply as a sheet for potential contributors of software. As soon as Eric decided to include news, reviews and tips, it became a kind of magazine. Nine bi-monthly issues have now appeared, a total of over a hundred tightly-typed A4 duplicated pages.

"The Newsletter has a life of its own," said Eric. "I sometimes wish it had become a real magazine like *Interface* for instance. It's not of course profit-making, for the subscriptions barely cover the costs. MUSE paid for it when it first appeared, for it was free then, and of course they remain ready to back it if necessary."



***Students as Arbourthorne Middle School, Sheffield, are lucky. They've got access to a number of ZX81's, thanks to one of their teachers, Steve Dommett, who has developed a number of educational programs available from the company EdZX (16 Grasmere Road, Dronfield Woodhouse, Sheffield S18 5PS). The school has three self-contained mobile units, 16K ZX81s with TV and cassette. One of them has a printer. A single large TV can be connected to any computer for class work. The three computers are together in this photo for a programming group. Normally they are spread around the school in classrooms.***

The Newsletter is read by many non-teachers as it is of fairly wide interest while remaining biassed towards learning needs. The bulk of it is news and reviews (relevant software, hardware and publications) but there are plenty of tips, program listings, calls for help and notices. Eric's attempt to collate all his notes about suppliers also led to the *EZUG Directory*, another immensely popular publication. "But again not a profit-making one," observed Eric wryly.

Tim then asked about the Spectrum and EZUG's plans for it.

"The Spectrum is a beautiful machine, of course, but I'm rather disappointed in that it moved away from the trend towards real computing for every pupil. EZUG will of course service the Spectrum in the same way as the ZX81. Maybe the micro-drives will get us over that awful duplication problem."

"What about Sinclair support?" was the next question. It seems that Sinclair have fought shy of coming too close to EZUG. However they did set up an award scheme for educational software, administered by MUSE and EZUG, and this was very successful.

"Sinclair, like us, want their micros to be accepted in education for the valuable and effective machines they are. At the moment there are too many decision-makers trained on terminals who react against Sinclair's energy and innovativeness and attempt to ban the machines."

They can't succeed of course. Eric reckons that there are more Sinclairs in use in teaching in Britain than all other micros combined. He would like to think that EZUG has played a significant part in bringing that about.

For details of EZUG, send a stamped addressed envelope or international reply coupons to Eric Deeson at Highgate School, Birmingham B12 9DS, UK.

Information about MUSE can be obtained from the same source, or from Freepost Bromsgrove B61 0JT.

## O Level Physics

### Paul Holmes from Sutton Colefield reviews this revision program by SCISOFT

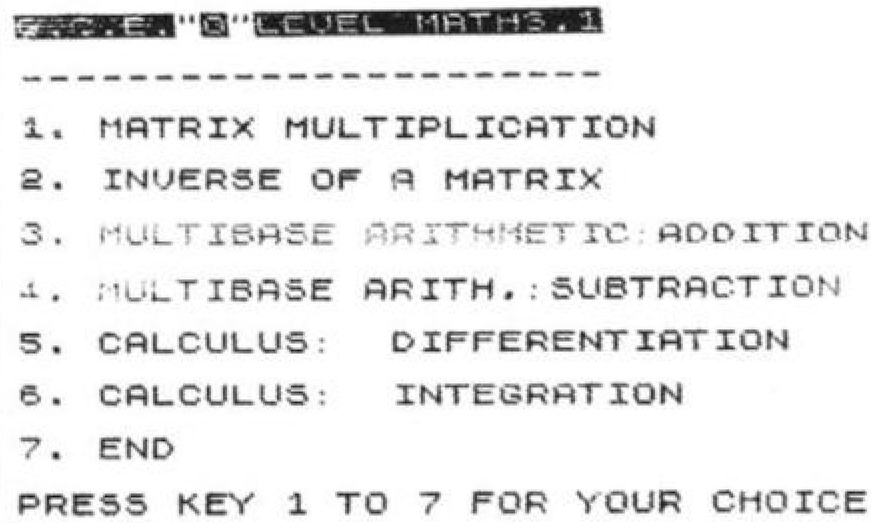
The revision package for O Level Physics from SCISOFT comes on cassette with a 30 page companion booklet for a price of £7.50.

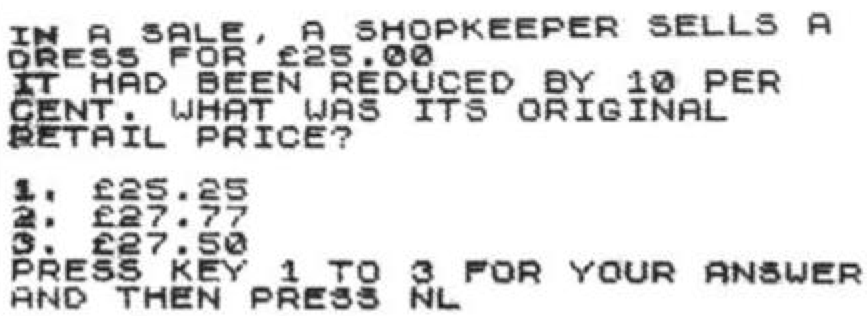
The booklet contains a brief introduction to the package giving instructions for use, but the main part is devoted to over 250 diagrams — a sensible idea in view of the ZX81's limited graphics. Each diagram is labelled and some have brief accompanying notes.

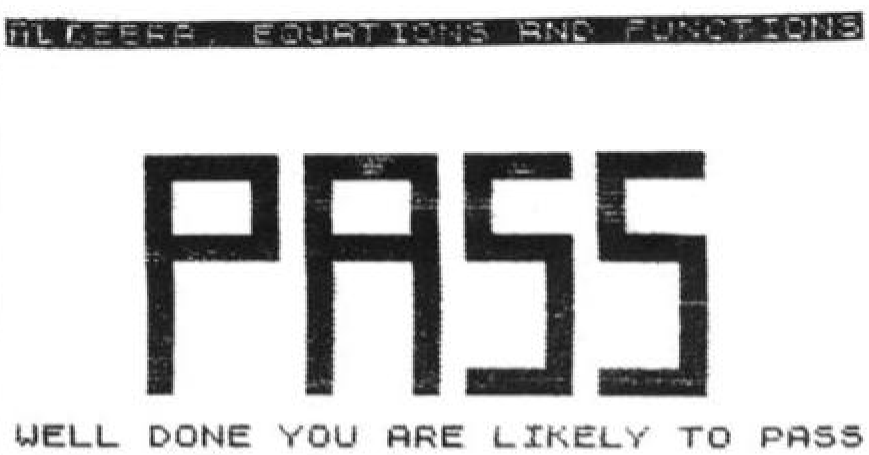
The cassette has eight programs, each using a full 16K which is an amazing 120K + of program power. The first program gives hints on revision and copes for those who are well organised, to those who are in a last minutes rush. For the later it portrays a gloomy outlook (it even draws a certificiate with FAIL and the user's name on it!!) with lines such as 'Panicking, eh?' and 'If this is the first time you have thought about revising then we suggest you try prayer'. The other seven programs are various tests and problems so you can determine where the grey areas are in your knowledge of the course.

I found the whole package a very good revision aid and as the program says, 'This will not pass your O Level for you', it is not all a tutorial but purely for revising. There are no explanations in the problems tests they are merely tell you the correct answer leaving it up to the user to find out how it should be done. Parts of the package are not quite as well finished off as I would have like them to have been. PAUSE was used for delays which gave a blink, inputs were not fully checked for illegitimate entries, and pointless moving graphics were used for the introduction and not to illustrate a point. One begins to feel he has his hands on Space Invaders, not doing a serious bit of revision.

Even in view of all this, at £7.50 it seems good value for money and helps you find out how much revision, and of what, must be done. Anyone considering buying it must consider the cost of other types of revision aids such as the 'Key facts cards' or the *LETTS revise* books by LETTS. These are all far cheaper than the cassette package but do not hold the advantage of being able to provide an infinite variety of tests. This is something that must be thought about before a purchase is made.




## Maths and Chemistry

### In Loughton, James Walsh turns reluctantly from studying to check out other O Level programs.

So you've come to the time when Mum and Dad think you've been spending too much time lately zapping aliens, killing monsters and basically defending the world from those phantom inverse asterisks and black blobs which are supposed to be the latest galaxian hyperspace-craft. It is about time you got on with the old school work. But wait! Next year when you're studying for the exams, you can tell Mum your latest program is actually teaching you.

As far as a lot of people are concerned an educational program consists of flashing a couple of random numbers up on the screen and asking for the answer. But to dispel this awful myth, I shall now look as three 16K cassettes written to help you pass O Level Chemistry and Maths. I'll start by looking at the Maths programs. The two cassettes I have for O Level Maths are: SCISOFT, Maths Part 1 (Part 2 not yet available) which costs £5.00; and Rose Cassettes, GCE O Level Maths, £4.50.

SCISOFT Maths comes complete with a 13 page manual. Only the first two pages are dedicated to the actual programs; the rest are revision notes. The cassette itself holds four 16K programs, all of which loaded the first time. The first program is called REVISION and starts off by asking you how long it is until your exams, as follows:

More than 6 months
3-6 months
2-3 months
One month
The Big Day

When you've picked your particular situation, it gives you ten pages of reasonably useful advice on how to revise, and in the case of The Big Day, helps prepare you for the examination itself. The minor bit I do object to is that on one frame it tells you that for a 'few' weeks prior to the exam you should have *no* social life whatsoever (sounds like a certain teacher I know). Apart from that, it gives a fair deal of very good advice.

The remaining three programs give you two questions on each of five basic question types. Though the questions are the same each run, the data is randomly generated. If you get the answer wrong, it just gives the correct answer without any explanation. The final screen (alias The Progress Report) shows you score, and a pretty representation of the word PASS, being overwritten to a certain degree by the word FAIL.

Though I feel that in some ways this is a good product, the computer isn't used to the fullest. For example, in the first program it would have been cheaper to put the information in the manual. I must admit I do like the idea of the pre-written notes. On minor factor which annoyed me was that everytime a key was pressed to change screens the screen flickered. This is due to the fact that PAUSE was used rather than a FOR/NEXT loop when the computer was to wait. OK, I thought, this must be to make it compatible with a new ROM ZX80; but no — it is advertised solely as a ZX81 product!

The second package I am going to look at is called GCE O Level Maths, developed and distributed by Rose Cassettes. Again, the whole lot is contained on one cassette, three programs in all. Unlike the SCISOFT cassette, all three have been recorded on both

sides in case one copy is damaged, but I found that all three loaded first time. The only literature which comes with this is a small leaflet with about seven and a bit lines on how to load the programs printed inside the cassette case. Personally, I don't think this is much of any problem as you don't really need much of an explanation anyway.

The first program is more of a lessons program, with a choice of six subjects. For each subject, you get screensful of explanation, one step-by-step example plus an exercise for you to do with random data so you can repeat it over and over again.

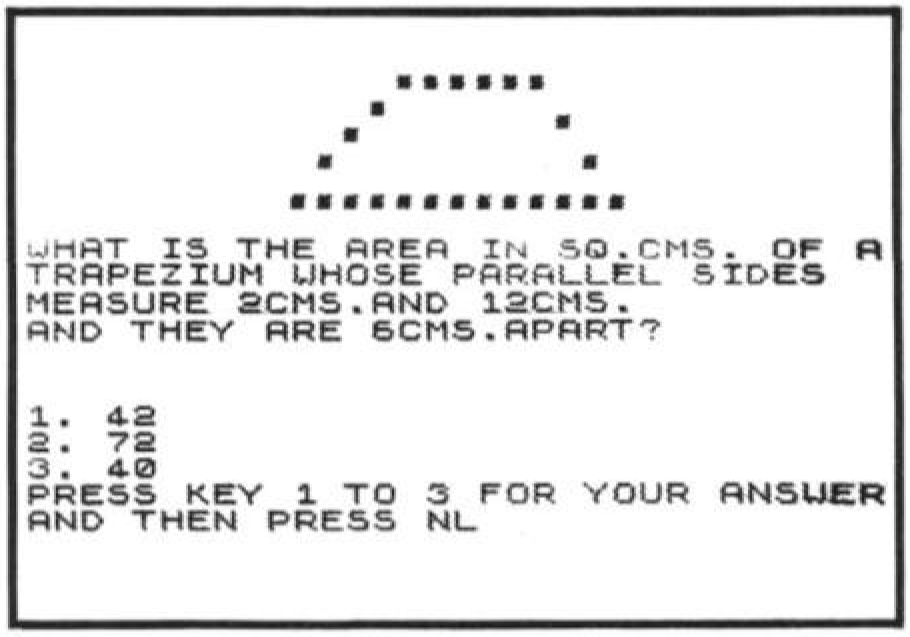
The examples and explanations are well-written and are quite enjoyable to use. The later two programs are timed multiple choice questions with ten questions per test. The nice thing about these are that not only do you have random data, but there is a choice of 30 different questions, rather than just the one.

## Comparing the two:

I feel that the Rose cassette makes far better use of the 16K RAM and concentrates far more on the questions and explanations than on pictures, which are hardly necessary. I liked the informal flavour of the Rose cassettes, as they do not talk down to you at all. So far as the acutal cassette is concerned, the Rose cassette is better produced than the SCISOFT one, but lacks any real instructions.

## Conclusion

Although both these cassettes are good value for money, I feel that the Rose cassette comes out better and at a lower price. I would like to see SCISOFT's Part 2 when it comes out as it may fill the gap Part 1 leaves behind.




# Not so boring after all

## James Walsh turns from aliens to alkalis and acids.

It is interesting to note that in may respects, educational programs have seemed to many people to have been the black sheep of computer programs. The idea of educational programs seems to bring most ZX owners to a state of the yawns. If this is your view (as it once was mine), then you are in for a pleasant surprise.

Subjects such as Chemistry and Biology don't really lend themselves to computerisation as easily as do subjects such as Maths, but the author of this program has got around the problem and made the whole process of learning or revising Chemistry more interesting . . . and almost fun.

The cassette holds two programs on each side, and each program needs 16K.

The title list reads:

1 - elements, compounds, mixtures an separations
2 - structure, bonding and properties
3 - redox, eletrolysis, and the activity series
4 - acids,bases and salts

For each program, the title will appear immediately after it has loaded, and ask you whether you want tutorial or test mode. In 'tutorial', it prints out two statements, and then asks you whether or not you think either one is true of false. If you decide that both statements are true, it then asks you whether the second one is a complete explanation of the first. The program will then tell you, one by one, whether or not you have the answer correct. At each stage it gives you the option of an explanation. It will give you ten pairs of statements for each program, and give you a score, as well as advice, at the end of each program. In 'test' mode, the same thing happens, but as if it were a test, without explanations.

I was studying for O Level Chemistry myself when reviewing this program, and found the questions interesting information and set out in such a way that I could actually enjoy going through them one by one. The explanations are concise and to the point and easy to understand. The whole set of programs is so well thought out, planned and produced that it is like a breath of fresh air in comparison to the numerous text books I have slogged through during the last few years.

I am very pleased to say that I can find very little apart from praise for this set of programs. There is only one matter which I would like all producers of education software to consider. If they want to penetrate the schools (which I presume they do), then it is necessary to enclose teachers' notes with programs. It may be true that the program, as in this case, only really needs the instructions on the inside of the cassette box, but most teachers would be daunted and discouraged by just a simple cassette (they haven't had the advantage of months of zaping aliens as we have had). We must remember that very few teachers, or many adults for that matter, really know much about computers. It is also important for the teacher to be able to plan how to use a computer program as part of their normal teaching plan. Though CALPAC do not supply any separate literature, I hope they will bear my suggestions in mind for the future.



***Jeff Warren, founder of CALPAC Computer Software, has been teaching for the past 12 years at the Farnborough College of Technology.***

## Conclusion

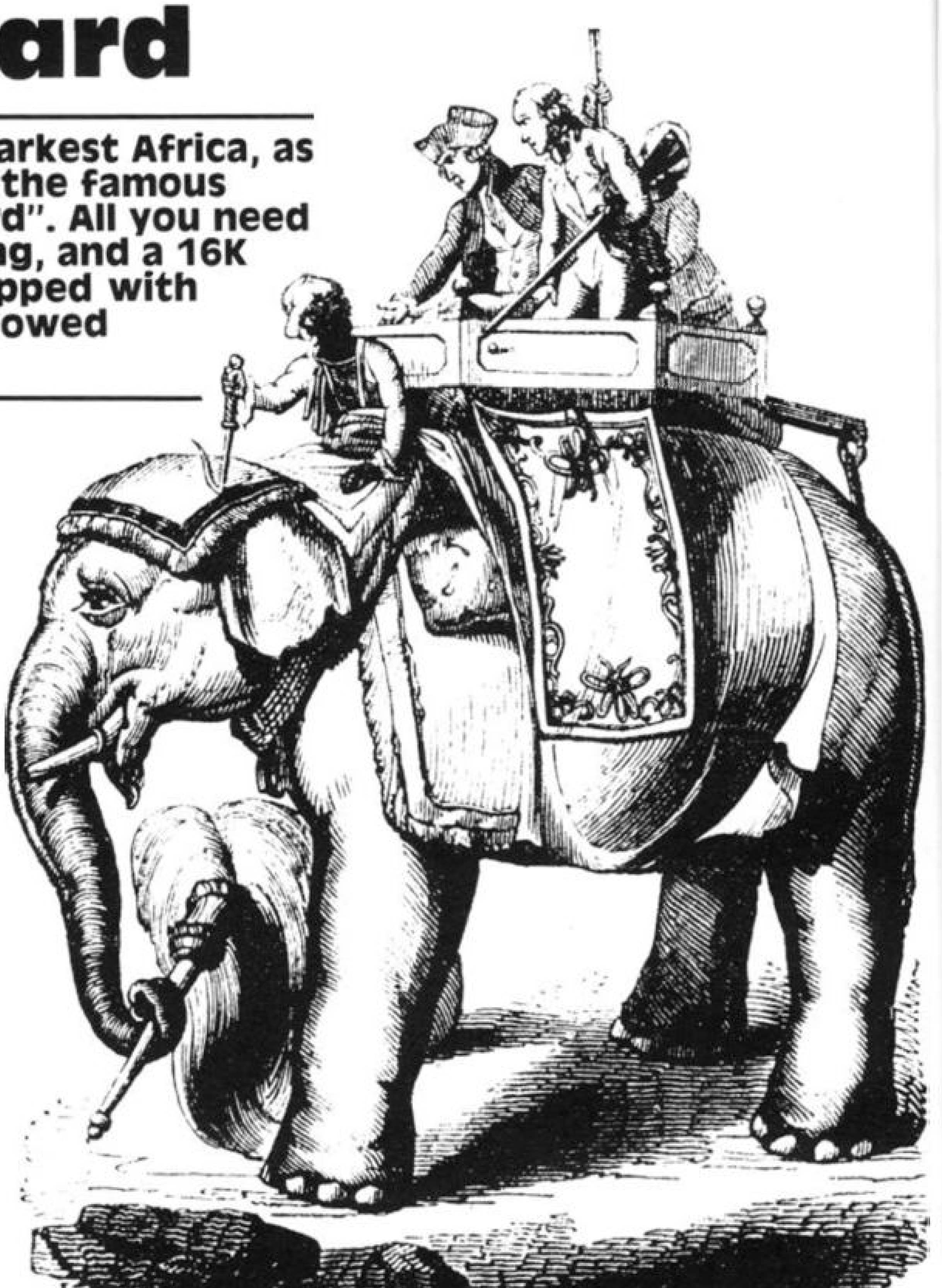
I would recommend this as an excellent supplement to the text book and as an invaluable revision aid to any fifth, fourth or particularly able third year student. O Level CSE Chemistry is available from CALPAC Computer Software, J. J. Warren, Hermitage Woods Crescent, St Johns, WOKING, Surrey GU21 1UF, for £4.95.

*If you're using ZX81s in your school let us know, and send us a photograph of the computer in use, naming everyone in the photo. Tell us what uses you've found for the ZX computers in educational situations so we can share your ideas with others.*

If you market educational software for ZX computers, we would like to review it in a forthcoming issue of *ZX Computing*. Our aim is to have the software reviewed by students actually studying the subject concerned, at the level for which the software is designed. This will ensure that the fairest and most useful review possible is achieved. Just send information on educational uses of the ZX computers, or software for review, to Education, *ZX Computing*, 145 Charing Cross Road, London, WC2H.

# The Elephant's graveyard

**Join Peter Shaw in Darkest Africa, as you attempt to find the famous "Elephant's Graveyard". All you need is your native cunning, and a 16K ZX81. Explorers equipped with ZX Spectra will be allowed to take part.**



You start the game with 100,000 KES (the local unit of currency), donated by generous people who believe the spirit of the old explorers is not yet dead. With this money you have to buy supplies, and hire natives to help you trek through the jungle and carry your supplies. Each of these assistants eats one food pack a week.

When you've bought your supplies, the game proper gets underway. The game lasts five rounds, with each round equal to one week. If you survive all five weeks, then you complete the game at the gates to the Elephant's Graveyard. If, however, you run out of money during the game, or food, or even natives, the screen goes black, and it is all over. There are a number of nasty surprises awaiting you within the jungle.

Variables used:
- A — KES
- C — five week loop
- N — natives
- F — food
- G — guns (with ammo)
- T — tents
- I — weekly pay for natives
- B, D, H, X, Y, Z — various inputs and loops
- A$ — various string inputs
- B$ — used in the electric storm subroutine

**Notes on program structure**

| Lines | |
|---|---|
| 120-290 | Ask how many of everything is wanted |
| 336-520 | Five week loop |
| 580-750 | Success routine |
| 1000-4030 | Hazard subroutines |
| 4500-4550 | Perfect week subroutine |
| 7000-7050 | Situation subroutine |
| 8500-8680 | Failure subroutine |

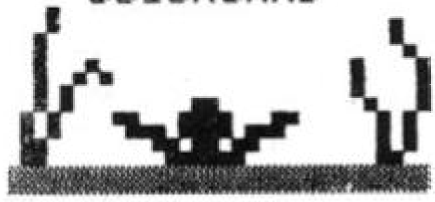
```
10 LET I=150
20 LET A=100000
25 LET Z=0
30 FOR B=1 TO 10
35 PRINT AT 0,8;"ADVENTURE INT
O"
40 PRINT AT 2,3;"THE ELEPHANTS
GRAVE YARD"
50 PRINT AT 0,8;"ADVENTURE INT
O"
```

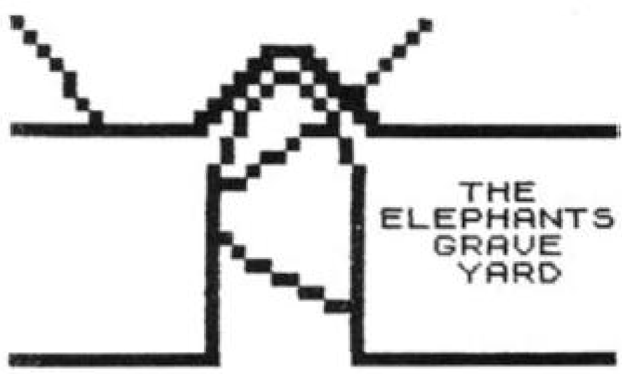
```
 60 PRINT AT 2,3;"THE ELEPHANTS
GRAVE YARD"
 70 NEXT B
 80 PRINT ,,,,TAB 6;"ANY KEY TO
CONTINUE"
 90 IF INKEY$="" THEN GOTO 90
110 CLS
120 PRINT TAB 10;"KES 100000"
125 PRINT ,,"HOW MANY NATIVES D
O YOU WANT TO HIRE AT 150 KES PE
R WEEK ?"
130 INPUT N
140 PRINT ,,"HOW MANY FOOD SUPP
LIES AT 50 KESEACH ?"
150 INPUT F
160 LET A=A-(F*50)
180 IF A<0 THEN PRINT "YOU ARE
OUT OF MONEY"
190 IF A<0 THEN GOTO 8500
200 PRINT ,,"HOW MANY GUNS (WIT
H AMMO) AT      1000 KES EACH ?"
210 INPUT G
220 LET A=A-(G*1000)
230 IF A<0 THEN GOTO 180
240 PRINT ,,"HOW MANY TENTS AT
500 KES EACH ?"
250 INPUT T
260 LET A=A-(T*500)
270 IF A<0 THEN GOTO 180
280 CLS
290 GOSUB 7000
330 PRINT ,,,,"PRESS ANY KEY TO
BEGIN"
335 IF INKEY$="" THEN GOTO 335
336 FOR C=1 TO 5
340 LET A=A-(N*I)
342 LET F=F-N
345 LET Z=0
346 IF A<0 THEN GOTO 180
347 IF F<0 THEN PRINT "YOU ARE
OUT OF FOOD"
348 IF F<0 THEN GOTO 8500
350 IF RND>.8 THEN GOSUB 1000
360 IF RND>.8 THEN GOSUB 1500
370 IF RND>.8 THEN GOSUB 2000
380 IF RND>.8 THEN GOSUB 2500
390 IF RND>.8 THEN GOSUB 3000
400 IF RND>.8 THEN GOSUB 3500
450 IF RND>.95 THEN GOTO 4000
460 IF Z=0 THEN GOSUB 4500
470 CLS
480 GOSUB 7000
490 PRINT ,,,,"ANY KEY TO CONTI
NUE"
500 IF INKEY$="" THEN GOTO 500
520 NEXT C
570 CLS
580 PRINT " ";TAB 15;" "
590 PRINT " ";TAB 9;"        "
600 PRINT TAB 2;" ";TAB 8;"     
 "
610 PRINT TAB 2;" ";TAB 7;"   
  "
620 PRINT "                          
   "
630 PRINT TAB 7;"          "
640 PRINT TAB 7;"          THE"
650 PRINT TAB 7;"          ELEPHAN
TS"
660 PRINT TAB 7;"          GRAVE
 "
670 PRINT TAB 7;"          YARD
 "
680 PRINT TAB 7;"          "
690 PRINT TAB 7;"          "
700 PRINT "                          
   "
710 PRINT ,,,,,,"YOU/VE DONE IT
"
720 PRINT ,,"WITH ";A;"KES TO S
PARE"
750 GOTO 8630
1000 LET Z=1
1020 LET H=INT (RND*10)+1
1030 CLS
1040 PRINT ,,"YOU ARE BEING ATTA
CKED BY LIONS"
1050 PRINT ,,"AND YOU HAVE ";G;"
GUNS "
1055 IF G<1 THEN GOTO 1200
1060 PRINT ,,"HOW MANY DO WANT T
O USE ?"
1070 INPUT Y
1075 IF Y>G THEN GOTO 1070
1080 IF Y*RND<.8 THEN GOTO 1200
1085 LET G=G-Y
1090 PRINT ,,"YOU WON,USING ";(Y
*2);" BOXES OF"
1100 PRINT ,,"AMMO"
1110 PRINT ,,,,"ANY KEY TO CONTI
NUE"
1120 IF INKEY$="" THEN GOTO 1120
1130 RETURN
1200 PRINT H;" OF YOU NATIVES WE
RE SAVAGED"
1210 LET N=N-H
1220 IF N<0 THEN PRINT ,,"YOU HA
VE NO NATIVES LEFT"
1230 IF N<0 THEN GOTO 8500
1240 PAUSE 200
1250 RETURN
1500 LET Z=1
1520 LET H=INT (RND*20)+1
1530 CLS
1531 FOR B=1 TO 7
1535 PRINT AT 0,0;
1540 PRINT TAB 3;"QUICKSAND"
1550 PRINT " ";TAB 14;" "
1560 PRINT " ";TAB 14;" "
1570 PRINT " ";TAB 13;"   "
1580 PRINT " ";TAB 13;"   "
1590 PRINT " ";TAB 14;" "
1600 PRINT " ";TAB 14;" "
1610 PRINT "                        "
1620 PRINT AT B-1,7;"  "
1630 PRINT AT B,4;"   "
1640 IF B+1<8 THEN PRINT AT B+1,
4;"   "
1650 IF B+2<8 THEN PRINT AT B+2,
5;"   "
1660 IF B+3<8 THEN PRINT AT B+3,
5;"  "
1670 IF B+4<8 THEN PRINT AT B+4,
6;"  "
1680 IF B+5<8 THEN PRINT AT B+5,
5;"   "
1690 IF B+6<8 THEN PRINT AT B+6,
4;"   "
1700 NEXT B
1710 PRINT ,,"YOU LOST ";H;" NAT
IVES IN THE"
1720 PRINT ,,"QUICKSAND"
1730 LET N=N-H
1740 IF N<0 THEN GOTO 1220
1750 PRINT ,,"ANY KEY TO CONTINU
E"
1760 IF INKEY$="" THEN GOTO 1760
1770 RETURN
2000 LET Z=1
2020 LET H=INT (RND*5)+1
2040 CLS
2050 PRINT ,,"YOU CAN SEE AN ELE
CTRIC STORM"
2060 PRINT ,,"COMING, DO YOU -"
2070 PRINT ,,"(A) SET CAMP AND W
AIT UNTIL ITS OVER"
2089 PRINT ,,"(B) GO UNDER A TRE
E FOR SHELTER"
2090 PRINT ,,"(C) CARRY ON WALKI
NG"
2100 IF INKEY$="" THEN GOTO 2100
2110 IF INKEY$="A" THEN GOTO 220
0
2120 IF INKEY$="B" THEN GOTO 230
0
2130 IF INKEY$="C" THEN GOTO 240
0
2140 GOTO 2100
2200 PRINT ,,"THE STORM DESTROYE
D ";H;" TENTS"
```

```
2210 LET T=T-H
2220 IF T<0 THEN PRINT ,,"YOU HA
VE NO TENTS LEFT"
2230 IF T<0 THEN GOTO 8500
2240 PRINT ,,"ANY KEY TO CONTINU
E"
2250 IF INKEY$="" THEN GOTO 2250
2260 RETURN
2300 PRINT ,,"THE TREE WAS HIT B
Y LIGHTNING"
2310 PRINT ,,"KILLING ";H;" NATI
VES"
2320 LET N=N-H
2330 IF N<0 THEN GOTO 1220
2340 GOTO 2240
2400 LET R=INT (RND*4)+1
2410 IF R=1 THEN LET B$="CAVE"
2420 IF R=2 THEN LET B$="HEN HOU
SE"
2430 IF R=3 THEN LET B$="ABANDON
ED HOUSE"
2440 IF R=4 THEN LET B$="POT HOL
E"
2450 PRINT ,,"WHILE WALKING YOU
FOUND A"
2460 PRINT ,,B$;"AND YOU"
2470 PRINT ,,"WERE UNHARMED"
2480 GOTO 2240
2500 LET Z=1
2510 PRINT ,,"YOU ARE LOST, SO Y
OU MUST SEND",,,"OUT SCOUTING PA
RTIES TO FIND"
2520 PRINT ,,"THE TRAIL."
2550 PRINT ,,"YOU HAVE ";N;" NAT
IVES"
2555 PRINT ,,"HOW MANY PER PARTY
 ?"
2560 INPUT X
2570 PRINT ,,"HOW MANY PARTIES ?
"
2580 INPUT Y
2600 IF Y*X>N THEN GOTO 2580
2620 IF (RND*Y)>3 THEN GOTO 2650
2630 IF (RND*X)>4 THEN GOTO 2650
2640 PRINT ,,"YOU ARE LOST FOREV
ER."
2645 GOTO 8500
2650 PRINT ,,"YOU FOUND THE TRAI
L"
2660 GOTO 2240
3000 LET Z=1
3010 CLS
3030 PRINT ,,"A SPOKESMAN FOR TH
E NATIVES SAYS"
3035 PRINT ,,"HIS MEMBERS WANT M
ORE MONEY"
3040 PRINT ,,"OR THEY WILL LEAVE
"
3045 PRINT ,,"HOW MUCH MORE (EAC
H) CAN YOU ",,,"OFFER ?"
3050 INPUT Y
3060 IF (RND*Y)>3 THEN GOTO 3200
3100 PRINT ,,"THEY HAVE ALL LEFT
 YOU"
3150 GOTO 8500
3200 LET I=I+Y
3210 PRINT ,,"THE NATIVES HAVE A
CCEPTED"
3215 GOTO 2240
3500 LET Z=1
3510 CLS
3530 PRINT ,,"YOUR PATH IS BLOCK
ED BY SNAKES"
3540 PRINT ,,"WILL YOU USE GUNS
OR WILL YOU"
3545 PRINT ,,"WALK PAST? (G OR W
)"
3550 LET H=INT (RND*10)+1
3560 IF INKEY$="" THEN GOTO 3560
3570 IF INKEY$="G" THEN GOTO 365
0
3580 PRINT ,,H;" OF YOUR NATIVES
 WERE KILLED"
3590 LET N=N-H
3600 IF N<0 THEN GOTO 1220
3610 RETURN
3650 PRINT ,,"YOU USED ";INT (H/
5);" GUNS"
3660 LET G=G-INT (H/5)
3670 IF G<0 THEN PRINT "YOU ARE
OUT OF GUNS"
3680 IF G<0 THEN GOTO 8500
3690 GOTO 2240
4000 LET Z=1
4010 CLS
4020 PRINT ,,"THE NATIVES HAVE R
EVOLTED AND",,,"LEFT YOU "
4030 GOTO 8500
4500 CLS
4510 PRINT ,,"PERFECT WEEK"
4520 PRINT ,,"YOU HAVE HAD A PER
FECT WEEK"
4525 LET H=INT (RND*100)+1
4526 LET F=F+H
4530 PRINT ,,"BONUS +";H;" FOOD
PACKS"
4550 GOTO 2240
7000 PRINT TAB 10;"KES ";A
7010 PRINT ,,"NATIVES ";N
7020 PRINT ,,"FOOD    ";F
7030 PRINT ,,"GUNS    ";G
7040 PRINT ,,"TENTS   ";T
7050 RETURN
8500 PAUSE 100
8510 CLS
8520 FOR A=0 TO 21
8530 PRINT "                                "
8590 NEXT A
8600 FOR A=1 TO 8
8605 PRINT AT 8,A+1;"MISSION FAI
LED"
8610 PRINT AT 8,A;"MISSION FAIL
ED"
8620 NEXT A
8650 PRINT AT 20,11;"ANOTHER GO"
8665 CLS
8660 INPUT A$
8670 IF A$="Y" THEN RUN
8680 STOP
```



QUICKSAND





From the Publishers of:




**Machine code and better Basic**
Ian Stewart & Robin Jones
with an educational program by
Eric Deeson

**£7.50** ISBN 0 906812 18 6 *July*

**The ZX81 Add-On Book**
Martin Wren-Hilton

**£6.50** ISBN 0 906812 19 4 *July*

**Easy Programming for the ZX Spectrum**
Ian Stewart & Robin Jones

**£4.95 (approx)** ISBN 0 906812 23 2 *August*

**Further Programming for the ZX Spectrum**
Ian Stewart & Robin Jones

**£7.50 (approx)** ISBN 0 906812 24 0 *September*

**Easy Programming for the BBC Micro**
Eric Deeson

**£4.95 (approx)** ISBN 0 906812 21 6 *September*

**Further Programming for the BBC Micro**
Alan Thomas

**£7.50 (approx)** ISBN 0 906812 20 8 *September*

## Shiva Publishing Limited

4 Church Lane, Nantwich, Cheshire CW5 5RQ
Telephone: (0270) 628272

Please supply me with ____________

Cheques should be made payable to Shiva Publishing Ltd

NAME (Capitals please) ____________

Full postal address ____________

For payment by Access/American Express:

Card No. ____________

Signature ____________

# ZX81 16K Spectrum

## MANAGEMENT GAMES

### AIRLINE

Can you compete with British Airways? You must decide on the number of aircraft to operate, whether to buy or charter, whether to enter into a loan or a fuel contract and the levels of staffing and maintenance. Problems encountered are tax demands, strikes, cancelled flighrs, hijacks and aircraft crashes.

### AUTOCHEF

As MD of a Catering Company you must negotiate for leases, decide on menu prices, level of wages, advertising and dividends. Each year you must predict the inflation rate. You are also given options on consignments of wines and food and loan contracts. You will be made to resign if you are not successful. There are 3 levels of difficulty.

### PRINT SHOP

You own a small printing company and are required to decide on (a) the number and type of staff you employ and when to increase or reduce staff (b) the amount and type of paper you stock (c) the week in which work is scheduled (d) the quotation for each job (e) cash requirements from the Cash Flow Statement. Are you an entrepreneur? Test your business acumen to the limit! There are 3 levels of difficulty.

These simulations are realistic and are not only fun to play but are educational. The user will learn to interpret Trading Profit and Loss Accounts and Balance Sheets. Our programs are very comprehensive and utilise over 15K.
ALL PROGRAMS INCLUDE DETAILED INSTRUCTIONS.

£4.75 for one, £8.00 for any two or £12 for three.
Please state computer type and send cheque to
**CASES COMPUTER SIMULATIONS**
**14 Langton Way**
**London SE3 7TL.**

# C.C.S.

NEW

# ZX 81 16K SOFTWARE

# DIGGLES KITCHEN



SIMPLE SUPPERS TO CELEBRATION DINNERS

## VOLUME 1

50 PAGES OF WORLDWIDE RECIPES
£4.99 (inc. P&P and VAT)

## VOLUME 2

50 PAGES EUROPEAN RECIPES
£4.99 (inc. P&P and VAT)

**Special price for two volumes**
**£9 (inc. P&P and VAT)**

More volumes to follow
Please specify which volume(s) – Mail order only
Send remittance to:-

**MICRO COMPUTER SOFTWARE**
**Unit D6, Pear Industrial Estate, Stockport Road, Lower Bredbury, Stockport SK6 2BP. Tel: 061-494 2441**

# Converting from other BASICs

**A wealth of computer programs written in BASIC can be found in a variety of books and computer programs, but as all versions of BASIC differ to some extent it is unlikely that a program written to run on another computer will work on the ZX81 or the ZX Spectrum without some changes. Dilwyn Jones of Bangor, Gwynedd, explains how to carry out the needed conversions.**

The extent and nature of the required changes depends greatly on the structure of a particular program and how it handles data, but it is possible to give some general guidance on things to look for when approaching the task of converting a 'foreign' program to run on a ZX81 or a ZX Spectrum. In the rest of this article, I'll refer to the ZX81, but my comments apply to the ZX Spectrum as well.

## (i) Multiple Statement Lines

Some BASICs allow multiple statements on a line, usually separated by : or \ , eg.

```
10 LET A = B(2)+C : PRINT
   A,B,C
```

These will have to be written on separate lines for the ZX81. Beware of multiple statement lines which involve IF...THEN conditional statements. In general, when an IF condition is false, control passes to the next line, not to the next statement. In other words, if the IF condition is false, the entire remainder of the line is skipped over. You should check that the BASIC does in fact operate in this way, and make allowances in your conversion attempts for this.

## (ii) Integers

The function INT on the ZX81 rounds *down* to the nearest integer. If the program requires that the number be rounded off to the *nearest* integer, then follow this procedure: If the number to be INT'ed is X, then to round off to the nearest integer used INT(X+0.5). Note that on the ZX81, the PRINT and PLOT commands round off to the nearest integer.

## (iii) Arrays

The first element of an array on the ZX81 is 1. In some BASICs, there is an additional subscript, 0, which is not available on the ZX81. Any program which uses the "zero subscript" must be altered to start at 1. One quick method (not always guaranteed to work) is to add one to each subscript value that you see used in the program. If this does not work, then the answer is to find out how the program works and rewrite the program so that the correct range of subscripts is obtained rather than modify the subscripts themselves.

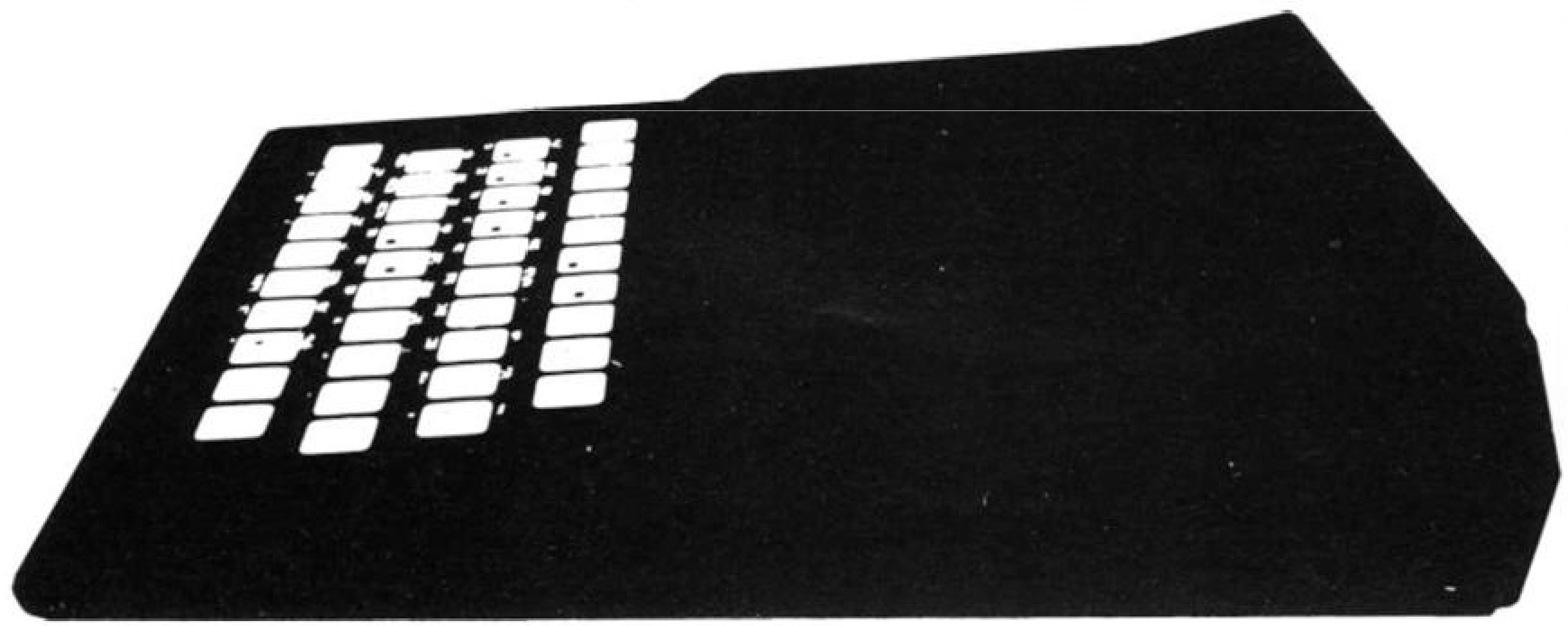
## (iv) LEFT$, RIGHT$, MID$

The string operator LEFT$(R$,X) may be replaced by R$(1 TO X) on the ZX81. This may be shortened to R$(1 TO X) on the ZX81. This may be shortened to R$(TO X), because 1 is the default value in this case. RIGHT$(R$,X) may be replaced by R$(LEN R$ − X + 1 TO LEN R$), which again may be shortened to R$(LEN R$ − X + 1 TO), because the default value in this case is LEN R$. MID$(R$, J,X) may be replaced by R$(J TO+K−1) on the ZX81.

## (v) LET

Some BASICs allow you to omit the LET word when assigning to a variable, but this is not permitted on the ZX81. Therefore if you come across say, 200 G = 88, then you must rewrite this as 200 LET G = 200.

## (vi) GOTO, GOSUB



Some BASICs do not allow a computed GOTO or computed GOSUB, such as GOTO BX30. It may, therefore, be possible to simplify a program using this ZX81 facility.

### (vii) ON . . . GOTO, ON. . . GOSUB

Often used in some basics, these statements are a form of computed GOTO/GOSUB. They make the program goto or gosub one of a number of lines depending on the value of the variable. For example, 55 ON A GOTO 115,220,333, which will jump to line 115 if I = 1, 220 if I = 2 or 333 if I = 3

The easiest way of converting this statement is by a series of IF. . . THEN GOTO lines, e.g.

```
IF A = 1 THEN GOTO 115
IF A = 2 THEN GOTO 220
IF A = 3 THEN GOTO 333
```

However, this is clumsy and wasteful of memory. If the line numbers increment neatly in fixed steps then it may be possible to use GOTO 500 + 30 x A for example (that is, make use of the computed GOTO/GOSUB facility). Note that this is not usually the case, but it is possible to sometimes renumber the program to suit. If the line numbers don't increment in convenient steps, then another possibility is to use 'GOTO a conditional expression'.

For example, ON A GOTO 115,220,333 could be replaced by GOTO (A = 1) x 115 + (A = 2) x 220 + (A = 3) x 333

Another possibility is:

```
GOTO (115 AND A = 1) +(220 AND A = 2) + (333 AND A = 3)
```

or even:

```
GOTO (115 OR A<>1) x (220 OR A<>2) x (333 OR A<>3)
```

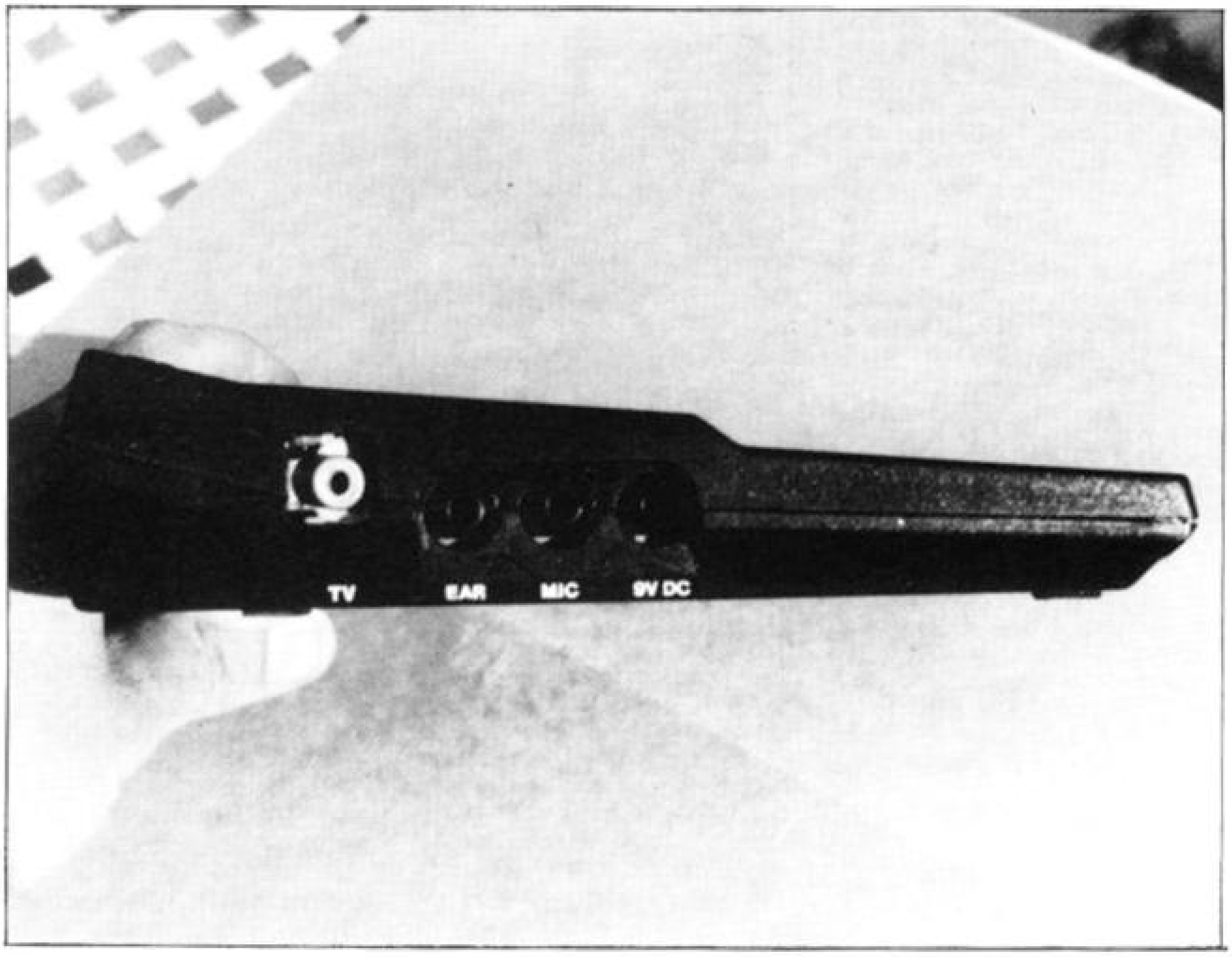
See "CONDITIONAL STATEMENTS" for an explanation of how these last three examples work.

### (viii) IF. . . THEN

The expression IF X = 2 THEN 200 is permitted in some BASICs. It means IF X = 2 THEN GOTO 200. You must include the GOTO after "THEN" the ZX81. Some BASICs insist on having a line number after THEN; the ZX81 can have any command after THEN; you may be able to use this facility to simplify programs on the ZX81.

### (ix) FOR...NEXT Loops

In many BASICs a FOR...NEXT




loop is executed at least once when it is met, even if the end value has already been exceeded, because the test to see if the end value has been exceeded is done at the NEXT statement. On the ZX81, if the end value has been exceeded before the loop starts, then the loop is totally and completely bypassed, eg.

```
FOR A = 1 TO 0
PRINT A
NEXT A
```

will result in nothing being printed, because the ZX81 had realised that 0 was less than the start value, so it decided to skip over the entire loop rather than run through it once. Note that if you added STEP −1, then the ZX81 would then perform the loop normally, because it then expects the finish value to be less than the start value. In general, this will not present problems unless the control variable is itself set by another variable.

Note also that the variable after NEXT may be omitted on some BASICs, in which case the most recent control variable is incremented. This is not possible on the ZX81, because the control variable must always be specified.

Some BASICs do not like you to jump out of a FOR...NEXT loop before that loop has been finished, and some require the use of a special statement enabling you to jump out of a loop. On the ZX81 you can jump out of a loop at will, although the control variable is stored in memory, meaning that you can jump back into that loop if you so desire. However, do not jump into a loop that has not already been executed, since this will cause the program to stop with an error report 1.

### (x) END

Sometimes may be omitted, sometimes may be replaced by STOP.

### (xi) PEAK and POKE

There is no easy way to convert statements involving these expressions, since their effect will be different on each machine. The only way to convert is to find out what the commands do, then rewrite the statement to perform an equivalent operation on the ZX81 if this is possible.

### (xii) INPUT

You may come across INPUT statements which can accept more than one input value, and perhaps print a prompt string as well. You will have to rewrite this using a PRINT statement for the prompt string, and a separate INPUT for each value required as data.

### (xiii) PRINT

It is highly unlikely that the PRINT format of the computer for which the program was intended will be the same as that of the ZX81. In certain cases, this will not matter, but if say, a moving display is required, or a line width exceeds 32 characters, then you may be in trouble. In cases where the spacing across the screen is merely to "look pretty", then you can easily change this by altering the TAB spacing or adding or omitting spaces in the PRINT statement. Note that programs designed to run on a printer or a screen larger than that of the ZX81 may need changing to prevent a display area overflow. One way of doing this is to have a subroutine to the effect of IF PEEK 16442 < 4 THEN CLS. This works because 16442 stores the line number of the PRINT position on the screen. If the subroutine discovers that the PRINT position has moved onto the bottom line, or whatever line you insert in the subroutine, then the screen is cleared automatically. Programs written for a printer can often be

CODE values) then use this routine:

```
1000 LET A$ = INKEY$
1010 IF A$ = "" THEN
     GOTO 1000
1020 LET A = CODE A$
```

Slightly different is the version that returns a numeric value rather than a character code. It is necessary to ensure that the character read from the keyboard is in the range "0" to "9" so that we can apply VAL to convert the character to a number. Here's one way:

```
1000 LET A$ = INKEY$
1010 IF A$<"0" OR A$>
     "9" THEN GOTO
     1000
1020 LET A = VAL A$
```

You may also come across a version of INKEY$ which allows a time limit to be specified for an user response, eg. 100 LET A$ = INKEY$(X) where X specifies the time limit. This can be converted in 2 ways:

First,

```
100  PAUSE X
110  LET A$ = INKEY$
```

and second,

```
100  FOR A = TO X
110  LET A$ = INKEY$
120  IF A$ <>"" THEN
     GOTO 131
130  NEXT A
```

You will need to fiddle the value of X for both routines to give the required time delay.

## (xxiii) VAL

If the argument of VAL does not form a valid numerical argument, you get an error report. Other BASICs return 0. See also IDIOT PROOF INPUTS.

## (xxiv) SET, RESET

These are used to make a particular screen point white or black. Replace with a PLOT/UNPLOT/PRINT AT.

## (xxv) DRAWING DIAGONAL LINES ON SCREEN

Some BASICs have a function that draws a line between two given sets of co-ordinates. The straightness and smoothness of this line is determined by the resolution with which the machine used can PLOT or PRINT the line. As the ZX81 does not sport high resolution graphics, and PLOTs on a 64 by 44 matrix, the lines produced are not impressive compared with a more expensive high resolution machine.

This little routine allows you to draw lines through two




less. You can use this routine to PLOT or PRINT one pixel or character if you want, simply enter the same pair of co-ordinates twice when prompted. The routine takes less than 300 bytes for program and variables, extra for screen. given sets of points. It may use PLOT or PRINT AT, and instructions are given to enable you to use either. You enter the co-ordinates in the following order:

(1) X co-ordinate you wish to start drawing *from*;
(2) Y co-ordinate you wish to start drawing *from*;
(3) X co-ordinate you wish to draw *to*;
(4) Y co-ordinate you wish to draw *to*;
For instance if you entered
0 NEWLINE
0 NEWLINE
63 NEWLINE
43 NEWLINE

you would see a line being PLOTted from the bottom left side of the screen up towards the top right side of the screen. It is quite fast to execute, the longest time to PLOT any line is seven seconds and the longest time to PRINT any line is four seconds. This applies to lines drawn across the full width of the screen: shorter lines take correspondingly Here is the routine:

```
8010 INPUT X
8020 INPUT Y
8030 INPUT X1
8040 INPUT Y1
8050 LET A = X - X1
8060 LET B = Y - Y1
8070 LET C = (A AND ABS
     A> = ABS B) + (B
     AND ABS B > ABS A)
8080 IF C = 0 THEN LET C
     = 0.1
8090 FOR F = 0 TO C
     STEP SGN C
8100 PLOT X+A/C * - F,
     Y+B/C * - F
8110 NEXT F
```

To PRINT AT rather than PLOT the line change line

```
8100 like this:
8100 PRINT AT Y+B/C * -
     F, X+A/C * - F
```

The INPUTs are not idiot-proofed at the moment, that is you can enter values which cause the program to crash or produce undesirable results. You may like to modify the program yourself to protect it against you and others. You can take one of two paths to do this. You can check each INPUT after it has been entered with a line like IF X<0 OR X>63 THEN GOTO 8010. Alternatively you can modify the loop to PRINT or PLOT only pixels or characters if their locations are actually on screen and ignore those co-ordinates that are off screen. Remember this is a subroutine rather than a program in itself although you can use it as a program if you add a line
8120 GOTO 8010

This will allow you to draw all sorts of lines to demonstrate the routine. Try drawing a frame around the screen, and lines from corner to corner. Experiment with the PRINT AT and PLOT version and see what they can both do. If you want anything other than black lines (eg. letters) then you'll have to use PRINT AT obviously.

## (xxvi) ELSE

This is an extension to the IF...THEN conditional statement and allows more than one outcome depending on whether the conditional statement is true or false. It may be replaced by two conditional expressions on the ZX81. For example

```
20   IF X = 1 THEN LET Y
     = 7 ELSE GOTO 80
```

may be replaced by

```
20   IF X = 1 THEN LET Y
     = 7
21   IF X <>1 THEN
     GOTO 80
```

If the action of ELSE is to assign one of several alternative values to a variable then it can be replaced on one line, eg.

```
50   IF X = 1 THEN LET Y
     = 7 ELSE LET Y = 8
```

may be replaced by

```
50   LET Y = (7 AND X =
     1) + (8 AND <>1)
```

Certain expressions such as the one above may be replaced by even shorter forms such as:

```
50   LET Y = 7 + (1 AND
     X<>1)
```

No general guideline can be given since the method used will vary from example to example — the examples above give an idea of what to expect.

You may come across a statement where the action performed by ELSE is itself conditional:

```
10   IF X = 1 THEN LET Y
     = 1 ELSE IF X = 5
     THEN GOTO 100
```

This will need to be rewritten as either:

```
10   IF X = 1 THEN LET Y
     = 1
11   IF X<>1 THEN IF X
     = 5 GOTO 100
```

or:

```
10   IF X = 1 THEN LET Y
     = 1
11   IF X<>1 AND X = 5
     THEN GOTO 100
```

Again, you may meet all sorts of conditional ELSEs, and the ZX81 versions will depend on the variation encountered.

## (xxvii) REPEAT... UNTIL

This is a loop that performs an operation continuously, ending only when a specified condition is met. Its use is so wide it is difficult to specify a universal method of conversion to ZX81 BASIC, probably the best being the IF...THEN GOTO conditional statement. Here is an example:

```
10   PRINT "ENTER YES
     OR NO"
20   REPEAT
30   INPUT A$
40   UNTIL A$ = "YES"
     OR A$ = "NO"
```

may be replaced by:

```
10   PRINT "ENTER YES
     OR NO"
20   INPUT A$
30   IF A$<>"YES" AND
     A$<>"NO" THEN
     GOTO 20
```

REPEAT...UNTIL structures are generally far more complex

than this example, and it may be necessary to find a means of conversion other than IF...THEN GOTO. For example, where the value of a variable is the determining factor, a FOR/NEXT loop may sometimes be used. However, the possibility of using the IF...THEN GOTO conditional statement should always be considered and is sometimes the only acceptable method of conversion

## (xxviii) UNDEFINED VARIABLES

If you attempt to use a variable before it has been defined or assigned to in a program, then some computers will return a value of 0. You get an error report 2 on the ZX81 if the variable has not previously been assigned to. So all variables must have been assigned to when using programs on the ZX81 which use variables.

## (xxix) MATRICES

Some BASICs have matrix functions which perform operations on arrays. The ZX81 does not have these functions, so you will have to perform the operations on array elements individually, possibly by means of a loop.

```
10     DIM X(Y)
20     DIM P(Y)
30     MAT X = P
```

This particular example can be replaced by

```
10     LET N = 0
20     DIM X(Y)
30     DIM P(Y)
40     LET N = N+1
50     IF N <Y THEN GOTO
       40
```

## (xxx) PROC, ENDPROC

This is a method of using subroutines to do certain procedures in such a way that among other things makes programs and listings easier to understand and read (it is called structured programming by some). It enables subroutines to be used specifically to do certain things and it is like a subroutine in many ways, but with the important exception that it is called by a name rather than by its line numbers. Take this simple example, which prints the score on the screen:

```
100    PROCscore
.....
1000  DEF PROCscore
1010  PRINT"SCORE=";S
1020  ENDPROC
```

ENDPROC is in a way similar to RETURN in that the procedure comes to an end and the program resumes from the line after the one which called the procedure, in this case the line after line 100. The name of the procedure is not used in the ZX81 version, although it can be adapted for the purpose as the second example ZX81 version will show. The simplest method of conversion to ZX81 BASIC is for line 100 to GOSUB line 1000, possibly have a REM statement somewhere in the ZX81 subroutine to identify it, and end the subroutine with a RETURN command.

```
100    GOSUB 1000
.....
1000  REM SCORE
          SUBROUTINE
1010  PRINT "SCORE=";S
1020  RETURN
```

If you want to retain the procedure/subroutine naming facility you can use a variable of the same name as the PROC name assigned during the course of the program before the subroutine is called, and use this variable as the destination for the GOSUB command. You could include a REM statement in the subroutine to identify the subroutine and tie it up with the variable name used. It is useful to use inverse characters in these REM statements so that they stand out from the rest of the listing text. So you can make ZX81 programs seem fairly structured if you like that sort of thing!

```
50     LET SCORE = 1000
.....
100    GOSUB SCORE
.....
1000  REM SCORE
          SUBROUTINE
1010  PRINT "SCORE = ";S
1020  RETURN
```

Although PROCs may be complex, an ordinary subroutine is the best method of conversion to ZX81 BASIC using GOSUB/RETURN.

## (xxxi) INSTR(A$,B$)

This is a function that looks to see if there is a copy of B$ in A$, and if there is it tells you where the copy starts. For instance, if B$ was "PUT" and A$ was "COMPUTER" then the value of INSTR(A$,B$) would be 4 because the part of A$ which held the letters "PUT" started at the fourth element of A$. If the function does not find a copy of B$ in A$, then INSTR(A$,B$) has a value of 0. A special routine has to be written to provide this function on the ZX81. Here is one method of converting this function to run on the ZX81. See also the version described in IDIOT PROOF INPUTS.

```
1000  REM LET X =
          INSTR(A$,B$)
1010  LET Y = 0
1020  IF LEN A$ = 0 OR
          LEN B$ = 0 OR LEN
          B$>LEN A$THEN
          RETURN
1030  FOR Y = 1 TO LEN
          A$ - LEN B$+1
1040  IF A$(Y TO Y+LEN
          B$ - 1) = B$ THEN
          RETURN
1050  NEXT Y
1060  LET Y = 0
1070  RETURN
```

Note that if you want to detect whole words rather than just strings then you will have to examine A$ for spaces or punctuation marks that signify the start and end of words. The routine above just finds matching strings, so that if you wanted to find the word CAT in a phrase containing the word CATASTROPHIC, this would trigger on the first three letters of CATASTROPHIC. See IDIOT PROOF INPUTS for advice on this. However, users of INSTR usually have this problem so the program will cater for this anyway!



## (xxxii) MOD

MOD gives the *remainder* of a division, eg. 17 MOD 5 is 2. A MOD B is A − (INT (A/B) x B) on the ZX81. Note that TAB carries out its own MOD action (modulo 32) on the ZX81.

## (XXXiii) RETURN, ENTER

Used normally, these correspond to NEWLINE. However, the ASCII code is not the same as the ZX81 code where this is important.

## (xxxiv) CURSOR MOVEMENT

Certain programs may require the use of cursor control codes to backspace over text or move the PRINT position. Where the cursor movement is absolute, then a simple PRINT AT Y,X; may suffice. Screen formats vary greatly and since the ZX81 has one of the lowest resolution screens around (32 by 22 characters), displays may prohibit the use of the same cursor controls. Where cursor movement is relative (eg. backspace 1 character) the following may help: use the values contained in the system variables 16441 (PRINT column number) and 16442 (PRINT line number) to control the PRINT position. The values contained in these system variables do not correspond to the normal PRINT AT Y,X; values. The PRINT line number (16442) starts off at 24 for a Y co-ordinate of 0. The PRINT column number (16441) starts off at 33 for a X co-ordinate of 0. So to move the PRINT cursor (!) up one position we could use: PRINT AT 24 − PEEK 16442 − 1,33 − PEEK 16441;

To move the PRINT position one position to the right: PRINT AT 24 − PEEK 16442+1,33 − PEEK 16441+1;

And to move the PRINT position one position to the left (provided the last PRINT statement ended with a semi-colon this could be used to erase the last character printed!);

You could save all the hassle if you used a variable to control the PRINT position as you would in a moving graphics program.

modified for a SCROLLing display. The only facility on the ZX81 is for an upward scrolling display (although a machine code program can be written for the ZX81 to SCROLL downwards or SCROLL part of the display). Note that the lines in a scrolling display on the ZX81 are only as long as they need be, ie. they are not filled up with spaces as are the normal lines on screen with more than 3¼K of memory attached (according to the system variable RAMTOP), so you may encounter problems if you attempt to PEEK or POKE the display.

## (xiv) Exponentiation

Some BASICs use the symbols ∧ or ↑ to represent exponentiation; the ZX81 uses xx

## (xv) DEF, FNR

This is a user defined function, which is mainly a shorthand way of writing an expression. You could replace this by writing the expression out in full each time it was needed, or by having a subroutine to perform the required calculation. Another method which is not always guaranteed to work is to assign the required calculation to a string variable and use VAL to evaluate the expression. This works because VAL can evaluate any numeric expressions including variables and numeric functions, eg. if the original user defined function reads

```
500   DEF FNR(S) = INT
      (RND x S) + 1
.....
2050 LET X=FNR(7)
```

convert it to—

```
500   LET AS= "INT
      (RND x S) + 1"
.....
2040 LET S=7
2050 LET X=VAL A$
```

This performs the same duties as a subroutine might but you may find it easier to use this method when converting "foreign" BASICs. You will find that in certain applications it can be faster than a subroutine. Note that you can replace the S in line 500 with a number and use this as a routine to generate random numbers in which case you can omit line 2040. Who knows — subroutines may eventually become redundant!

## (xvi) Random Numbers

On machines dealing in real numbers, ie. machines which are capable of handling floating point numbers, random numbers are usually generated by the expression RND(0) or RND(1) or RND. The number yielded is usually between 0 (which value can be taken) and 1 (which value cannot be taken). This can be directly replaced by RND on the ZX81. On machines which handle only integer numbers random numbers are usually generated by the expression RND (X), which usually yields any number from 1 to X inclusive. The equivalent expression on the ZX81 is INT (RND x X) + 1, which yields an integer in the same range. Since individual BASICs do vary, ensure that the minimum value is 1 and not 0. If so, omit the +1 in the ZX81 expression. Remember that the method of obtaining the seed for the random numbers (if there is one) may well be different. For what's worth the ZX81's RAND function works as follows:

The number you place after the word RAND is stored in the system variables 16434 and 16435 after being rounded off to the nearest whole number. If this whole number is outside the range 0 to 65535 then error B results. If you just enter RAND or RAND 0 then RAND is given the value of the frame counter in addresses 16436 and 16437. This value is *not* affected by CLEAR or RUN, but is reset to 0 by NEW, as it is at switch-on. It changes every time you use RND.

## (xvii) ASC, CODE

ASC returns the ASCII (American Standard Code for Information Interchange) code of the first character in the string. It is similar to the ZX81 CODE function, except that the numbers yielded are different. There is no easy method to convert values (a table of ASCII codes is given elsewhere in the book) except to add 20 to the CODEs of numbers from 0 to 9 and add 27 to the ZX81 character CODE of any capital letters from A to Z, you will be given the ASCII code of that letter. Note that several ASCII characters, including lower case letters are not available on the ZX81.

## (xviii) READ,DATA RESTORE

More BASICs allow you to write a list of data elements in the program. When the program is RUN, a READ statement is then used to transfer the values to an array. The simplest way of converting is to replace the lot with a list of LET statements. This can be very tedious and consume a lot of memory if there are several values. A better method is to use the routine in the section PRINTING STRINGS elsewhere in this book. First declare an array with sufficient dimensions and enter the elements individually by means of a loop, then delete the initialisation program and save the rest of the program on tape using the load and go routine, to avoid any risk of starting the program with RUN, and deleting all your carefully preserved variables.

Another method is to set up a string array long enough to accommodate all the data in one string, then set up a numeric array so that the first element says where the first word or data element starts, the second says where it ends, the third indicates the start of the second word or data element, the fourth the end of that second data element and so on. Here's an example of this in use. The computer will achieve the amazingly difficult task of telling you which month your birthday falls in if you give it the number of that month.

You will need two arrays,



A$ and B$. A$ holds information concerning the location of words in B$. B$ may be up to 999 characters in length with three digit storage in A$. You will need to alter several things in the program to change the number of digits that store information in A$.

You also need a numeric variable A which tells the ZX81 which word you want to extract from the data string B$. If like, A is the number of the word you READ from the DATA string. There is no need for a RESTORE command since the variable A can simply be reset to 1 if you wanted to READ words from B$ in turn. You should include a line to preclude unwanted values of A (in this case, less than 1 or greater than 12) since these will constitute a subscript error and cause the program to STOP with error report 3. Here is the routine:

```
10 LET A$ = "001008016
   021026029033037043
   052059067075"
20 LET B$ = "JANUARY
   FEBRUARYMARCHAPRIL
   MAYJUNEJULYAUGUST
   SEPTEMBEROCTOBER
   NOVEMBERDECEMBER"
30 PRINT "ENTER THE
   NUMBER OF THE
   MONTH YOU WERE
   BORN IN?"
40 INPUT A
50 IF A< 1 OR A> 12
   THEN GOTO 40
60 LET A=(A-1) x 3
70 PRINT "SO YOU WERE
   BORN IN";B$(VAL A$ (A
   + 1 TO A + 3) TO VAL
   A$ (A + 4 TO A + 6)-1
```

The numbers in string A$ are arranged in groups of three to simplify decoding, for example the first three digits refer to the starting position of the first word, 001, the second set of three digits to the starting position of the second word, ie. 008 and so on. You may have noticed that there are an extra three digits at the end of A$ that refer to a non-existent element — in fact it is one greater than the position of the last character in B$ and is necessary for the correct functioning of the routine. This is because, to find the end of a word the routine looks for the beginning of the next word and subtracts one from its starting position. As it stands, the routine allows you to store up to 999 characters of DATA because the starting positions are stored as three digits which gives you a maximum number of 999. To store more DATA than this you need to store the information in A$ in 4 digits and change the decoding as necessary in lines 60 and 70. Remember that the maximum value of A allowed in line 50 should be the same as the number of words in B$. It may be less if you want to restrict the amount of words available, eg. anybody with a birthday later than OCTOBER was not allowed to use the program!



The routine runs fairly quickly, and if you want to test its speed, make the following changes to the routine:
delete line 30

```
40 LET A = INT(RND x 12)
   + 1
70 PRINT B$ (VAL A$ (A +
   1 TO A + 3) TO VAL A$
   (A + 4 TO A + 6)
   - 1);" ";
80 GOTO 40
```

What do you suppose happens if A is not a whole number? How could you prevent this happening? You could add a line like
45 LET A = INT A

See if you can improve this, possibly adding INT to an existing line.

## (xix) Integer Arithmetic

In general, always add the function INT before a division in a program designed for a computer with integer arithmetic. You may require brackets around the division so that INT works only on the result of that division.

## (xx) Logical Expressions

Most BASICs allow expressions to be evaluated as true or false. On the ZX81 a true expression returns a value of 1, a false returns a value of 0. Some BASICs return −1 for a true expression. The particular method of conversion used will depend on the context in which the expression is used. It may be possible to negate the result by simply adding the − symbol to the expression, eg. LET A = B = C may be replaced by LET A = − (B = C). This method will not work all the time and hence it may be necessary to completely rewrite the expression for it to work properly on the ZX81.

## (xxi) DIM

Some BASICs allow you to write several DIM statements on one line such as DIM A$(9),B$(8),C$(7). You will have to replace this by individual DIM statements on separate program lines. If the program calls for arrays with names that are more than one letter long, then these have to be replaced by single letter names like A$ or B. If you do not have enough letters available then you may be able to declare additional dimensions to the existing ones for a certain array and use the extra dimension to replace an array. Programs that cause this problem are generally too long to fit into a ZX81 anyhow. Beware of the zero subscripts!

## (xxii) GET, GET$

This is a function that reads characters or values from keys pressed on the keyboard. It takes various forms on various computers, but in general it waits until a key is pressed before it goes on, assigning either the character corresponding to the key pressed or the code of that character to a variable. For example, GET A$ or LET A$ = GET$. You could do this on the ZX81:

```
1000 LET A$ = INKEY$
1010 IF A$ = "" THEN
     GOTO 1000
```

This would return the character corresponding to the key pressed on the keyboard. If the function was to return the CODE of the character (NOTE: this would be ASCII code, which returns completely different values to the ZX81

# Dodgem Chomp

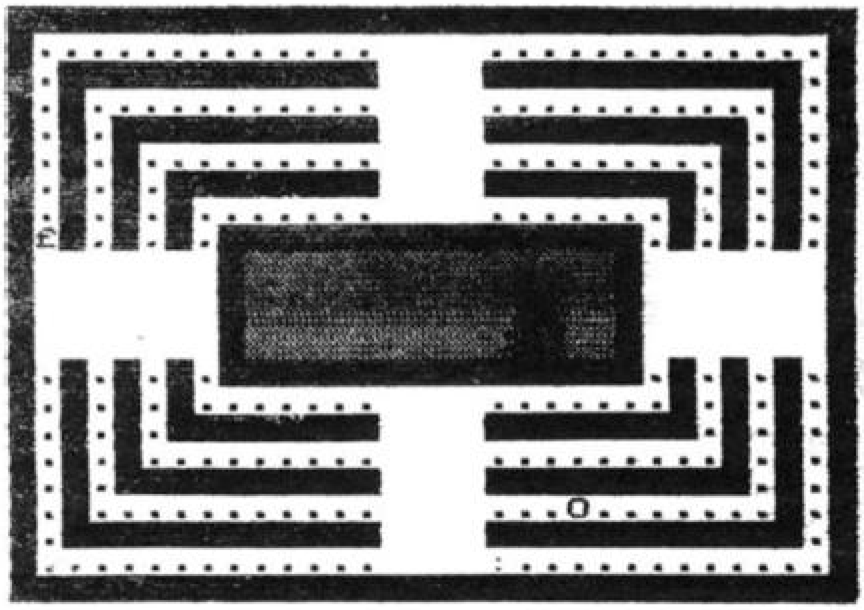
**Tim Rogers from Richmond turns his programming skill to create a 'Dodgem car' type program called CHOMP.**

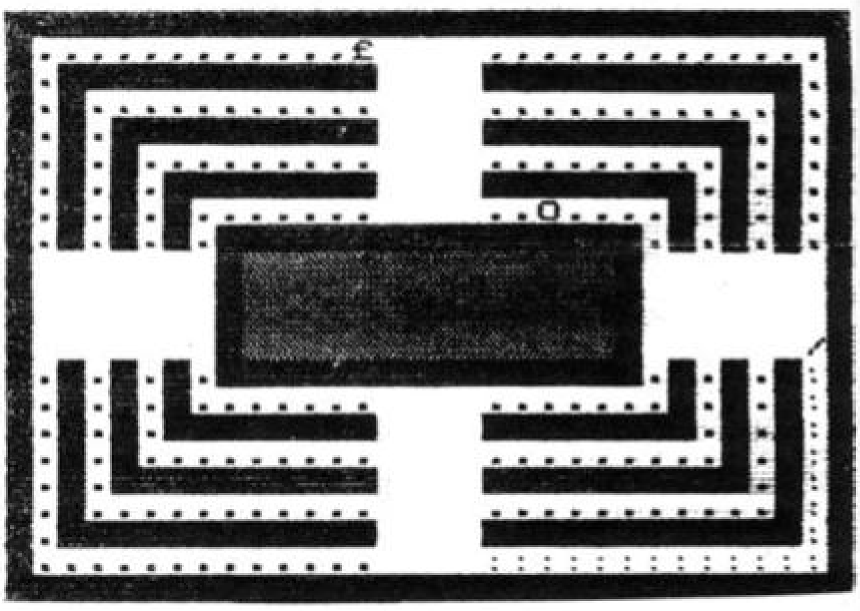
You are a hungry snake on an oblong course filled with dots of food. By typing in RUN you start the little creature chewing its way anti-clockwise round the course leaving crumbs in its wake. You cannot stop it or reverse its direction, but when you get to one of the four gaps you can change lanes by pressing one of the four arrow keys corresponding to the direction you want to go in. For example if you are heading north and want to change to a lane further on the inside press the left arrow key (key 5). However, life is not that simple for this hungry little snake as there is a rather nasty monster going in the opposite direction (clockwise round the course) which is determined to eat the snake. It bumps round at the same speed as the snake but has less manoeuvrability when it comes to changing lanes: Whereas the snake can change up to four lanes per gap, the monster can only change one lane.

The monster is always out to get you and will try to be in the same lane as you and thus cause a head-on collision and swallow you up. There will always be a 'barrel' somewhere on the course which the snake will swallow even more readily, but beware, once eaten, a space will remain which the monster will treat as a gap and, if necessary, use it to change lanes. Ordinary dots score one point each when eaten; barrels score five. The snake can go across crumbs where dots have already been eaten but it will not score points. Once all the dots have been eaten the snake will begin to eat crumbs leaving behind dots, and when all the crumbs have been eaten the snake will begin to eat dots again, although you have to be very good indeed to get to this stage.

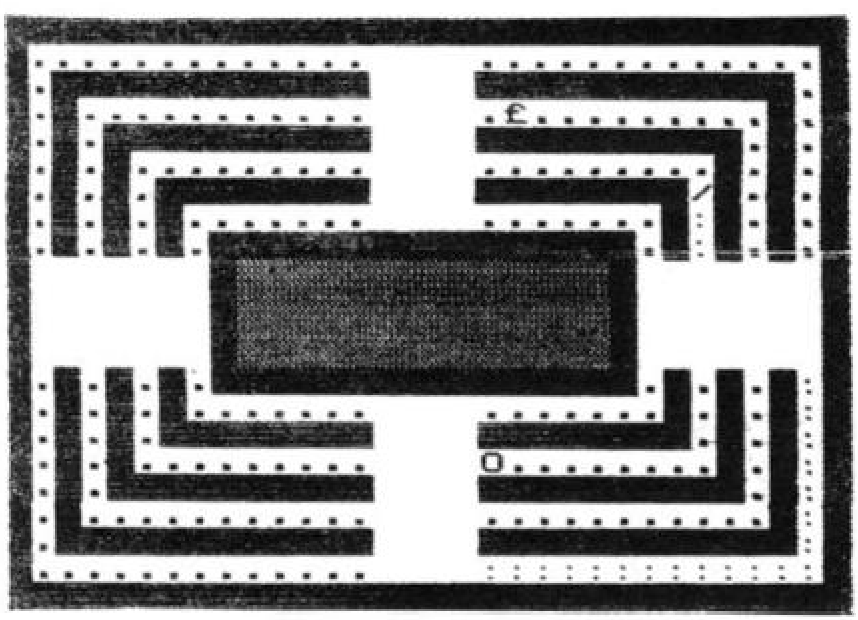
In theory the game can go on forever but because of the barrels Tim Rogers *thinks* that the highest possible score you could get is 44256 (5 × 208 + 208²). His record is about 450 or so which, with a little practice, could easily be beaten.

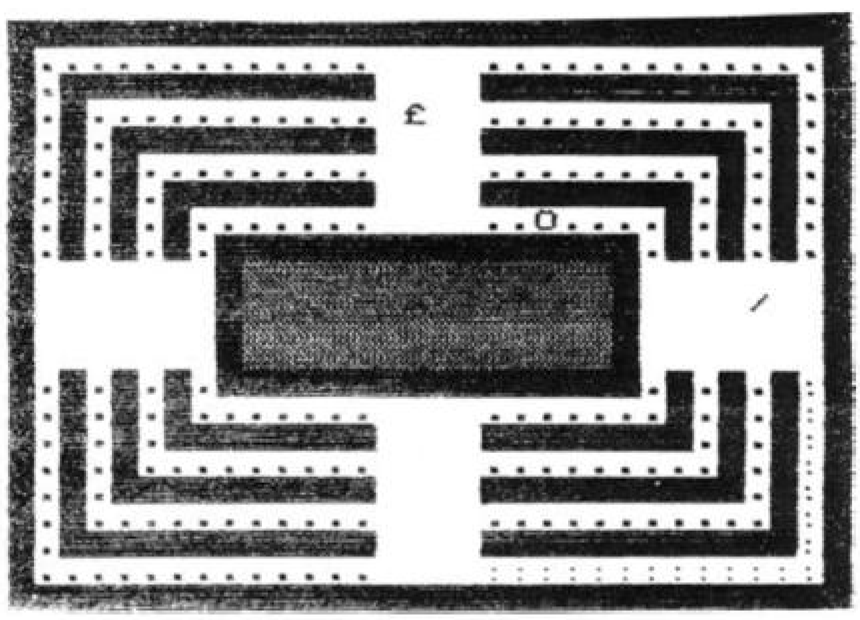
```
   1 REM ....................
   2 GOSUB 1000
   3 SLOW
   4 LET H1=0
   5 LET S=0
   6 LET S1=0
   8 RAND
  10 PRINT "████████████████████████████████"
  20 PRINT "█..............  .............█"
  25 LET Q=0
  30 PRINT "█.████████████  ███████████.█"
  40 PRINT "█.█............  ..........█.█"
  50 PRINT "█.█.██████████  ████████.█.█"
  60 PRINT "█.█.█..........  ........█.█.█"
  70 PRINT "█.█.█.████████  ██████.█.█.█"
  80 PRINT "█.█.█.█.......  .....█.█.█.█"
  90 PRINT "█.█.█.█.██████████████.█.█.█.█"
 100 FOR A=1 TO 4
 110 PRINT "█";TAB 8;"████████████████";TAB 31;"█"
 120 NEXT A
 130 PRINT "█.█.█.█.██████████████.█.█.█.█"
 140 PRINT "█.█.█.█.......  .....█.█.█.█"
 150 PRINT "█.█.█.████████  ██████.█.█.█"
 160 PRINT "█.█.█..........  ........█.█.█"
 170 PRINT "█.█.██████████  ████████.█.█"
 180 PRINT "█.█............  ..........█.█"
 190 PRINT "█.████████████  ███████████.█"
 200 PRINT "█..............  .............█"
 202 LET V=224
 205 LET G=14
 207 LET H=27
 209 LET V1=V
 210 PRINT "████████████████████████████████"
 215 LET A1=PEEK 16396+256*PEEK 16397
```





```
 220 LET B1=0
 225 LET LA=1
 230 LET A=A1+678
 235 LET LB=1
 240 LET B=A1+299
 245 LET A2=0
 250 LET C=1
 252 LET O=A1+INT (RND*660)
 254 IF PEEK O<>27 AND PEEK O<>1
4 THEN GOTO 252
 255 IF PEEK O=H THEN LET V1=V1-
1
 256 POKE O,52
 257 LET V=V-1
 258 IF S>1 THEN RETURN
 260 LET D=-33
 270 IF A2=H THEN LET S=S+1
 280 IF PEEK (A+C)=128 THEN GOSU
B 400
 290 POKE A,G*((A2=H)+(A2=G))
 292 IF S=V1 THEN GOSUB 900
 295 IF INKEY$<>"" AND PEEK (A+C
)=0 THEN GOSUB 700
 300 LET A=A+C
 301 IF A=O THEN LET S1=S1+5
 302 IF A=O THEN GOSUB 252
 304 LET A2=PEEK A
 305 IF PEEK A=12 THEN GOTO 500
 310 POKE A,24
 320 IF PEEK (B+D)=128 THEN GOSU
B 450
 330 POKE B,B1
 335 IF B1=0 AND Q=0 AND LA<>LB
THEN GOSUB 800
 337 IF B1<>0 THEN LET Q=0
 340 LET B=B+D
 345 IF PEEK B=24 THEN GOTO 500
 350 LET B1=PEEK B
 360 POKE B,12
 370 GOTO 270
 400 LET X=0
 402 IF C=1 THEN LET X=-33
 405 IF X=-33 THEN GOTO 435
 410 IF C=-33 THEN LET X=-1
 415 IF X=-1 THEN GOTO 435
 420 IF C=-1 THEN LET X=33
 425 IF X=33 THEN GOTO 435
 430 IF C=33 THEN LET X=1
 435 LET C=X
 440 RETURN
 450 LET Y=0
 452 IF D=-33 THEN LET Y=1
 455 IF Y=-33 THEN GOTO 485
 460 IF D=1 THEN LET Y=33
 465 IF Y=33 THEN GOTO 485
 470 IF D=33 THEN LET Y=-1
 475 IF Y=-1 THEN GOTO 485
 480 IF D=-1 THEN LET Y=-33
 485 LET D=Y
 490 RETURN
 500 POKE A,23
 510 FOR M=1 TO 26
 520 RAND USR 16514
 530 NEXT M
 585 LET S=S+S1
 590 SLOW
 600 PRINT AT 9,9;"SCORE▒";S
 605 IF H1<S THEN LET H1=S
 610 PRINT TAB 9;"HI-SCORE▒";H1
 620 PAUSE 35000
 630 CLS
 640 GOTO 5
 700 LET A3=A
 705 LET A$=INKEY$
 710 LET A=A+(((INKEY$="8")-(INK
EY$="5"))*(ABS C=33)+((INKEY$="6
")-(INKEY$="7"))*33*(ABS C=1))*2
 720 IF A>A1+726 OR A<A1 OR PEEK
A<>0 THEN LET A=A3
 730 IF A=A3 THEN RETURN
 740 LET L5=LA+(C=-1)*(A$="6")+(
C=1)*(A$="7")+(C=-33)*(A$="5")+(
C=33)*(A$="8")
 750 IF L5=LA THEN LET L5=LA-1
 755 LET LA=L5
 760 RETURN
 800 LET Q=1
 810 LET D1=D
 820 GOSUB 450
 830 LET D2=D
 840 LET D=D1
 850 LET W=LA-LB
 860 IF W>1 THEN LET W=1
 870 IF W<-1 THEN LET W=-1
 375 LET LB=LB+W
 880 LET B=B+W*D2*2
 890 RETURN
 900 LET S1=S1+S
 910 LET S=0
 920 LET G=H
 930 IF H<>G THEN GOTO 950
 940 LET H=14
 950 LET V1=V
 950 RETURN
1000 LET M$="042 012 064 006 023
 043 035 126 254 118 032 003 016
 246 201 198 126 119 024 242"
1010 FAST
1020 FOR M=16514 TO 16533
1030 POKE M,VAL M$( TO 3)
1040 LET M$=M$(5 TO )
1050 NEXT M
1060 RETURN
```





# Spectrum takes off

**The sound and colour on the Spectrum are two good reasons for buying one — and these three programs, one each by Alan Gunnell, Chuck Hopper and Anne Marshall show the sound and colour off to advantage.**

## Final circuit

Our first program — FINAL CIRCUIT — was adapted from a ZX80 program (2K RACETRACK) first published in the monthly magazine, *INTERFACE*.

It is easy to play, and because it ends up giving you a score after each 'race', acts as a challenge to play it over and over again, trying to increase your score. There are three 'racetracks' on which you can drive at varying degrees of difficulty.

Throughout the race, you are asked to enter your choice of acceleration and gear setting. You'll soon learn the effects these have. Your score is shown at all times (line 220), and a final score is given at the end. Your feedback (including such lines as 'Driver behind is hooting, hurry up' if you're dragging your heels) is in words, and appears throughout the race. You'll find there is a great tendancy to crash, and your vehicle manages somehow to survive an infinite number of crashes. Of particular interest is line 290, which takes the place of five IF/THEN statements of the type IF H = 5 THEN LET b$ = "oily straight" and so on.

```
   5 REM Final Circuit
  10 REM Adapted from ZX80
  12 REM program by Alan Gunnell
  14 REM First published in
  15 REM        INTERFACE
  18 REM
  20 LET a=5: LET g=1: LET b=3
  22 BORDER 1: PAPER 7: INK 0
  25 INPUT "Which track (3 TO 5)
?";v
  27 IF v<3 OR v>5 THEN GO TO 25
  30 LET x=0
  40 LET l=100+v*v
  50 LET s=0
  60 IF x=10 THEN STOP
  80 LET x=x+1
  90 IF x=10 THEN PRINT INK RND*
6;TAB 8;"THE RACE IS OVER"'TAB 4
;"Score is ";l;" out of ";100+v*
v: POKE 23692,-1: BORDER RND*7:
BEEP .02,RND*30: GO TO 90
 110 GO SUB 180
 112 FOR t=1 TO 50: BEEP .02,t:
NEXT t
 115 GO SUB 270
 120 PRINT INK RND*6;b$
 125 GO SUB 145
 130 GO SUB 350
 135 PAUSE 50
 140 GO SUB 270
 142 GO TO 60
 145 FOR t=1 TO 50: BORDER RND*7
 NEXT t: BORDER 1
 149 LET s=ABS (s+(a*a)-(b*15)+(
2*g))
 150 PRINT '''' PAPER 2; INK 6;"
Gear ";g;", speed ";s
 160 GO SUB h*1000
 180 BORDER 1: INPUT INK 7;"Sele
ct a gear (1 TO 10)";g
 190 IF g<1 OR g>10 THEN GO TO 1
80
 200 INPUT INK 7;"Enter accelera
tion (0 TO 10)";a
 210 IF a<1 OR a>10 THEN GO TO 2
00
 220 PRINT INK RND*6;"Current sc
ore is "; INK 2;l
 240 INPUT PAPER 2; INK 6;"Enter
 braking (0 TO 10)";b
 250 IF b<0 OR b>10 THEN GO TO 2
40
 260 RETURN
 270 LET h=INT (RND*v)+1
 290 LET b$=("oily straight" AND
 h=5)+("hairpin" AND h=4)+("corn
er" AND h=3)+("bend" AND h=2)+("
straight" AND h=1)
 340 RETURN
 350 IF a=0 THEN LET a=1
 360 LET s=ABS (s+(a*a)-(b*15)+(
2*g))
 370 IF s<10 THEN LET s=10
 380 IF s<15 THEN PRINT ''''' IN
K 2;"Driver behind is hooting":
PRINT ' INK 1;"Hurry up"
 400 RETURN
1000 IF s>90 THEN PRINT INK 2;"Y
ou're speeding...slow down!": LE
T l=l-5
1010 RETURN
2000 IF s>40 THEN BEEP 3,50: FOR
 q=1 TO 20: BORDER RND*6: NEXT q
: PRINT INK 2;"Crash"
2010 IF b>8 THEN PRINT INK 2;"Cr
ash": BEEP .5,20: BORDER RND*6:
PAUSE 20: LET l=l-9+INT (RND*10)
2020 RETURN
3000 IF s>25 THEN FOR r=1 TO 10:
 PRINT INK RND*6;"Crash!!!!!!!!!
": NEXT r: LET l=l-10
3010 RETURN
4000 IF s>35 THEN PRINT INK 2;"*
*****Crash*********************":
 LET l=l-10
4010 RETURN
5000 IF s>20 THEN PRINT INK 2;"C
CCCCrrrrraaaassssshhh!!!!!": LE
T l=l-10
5005 FOR t=1 TO 50: PAPER RND*7:
 CLS : NEXT t: PAPER 7
5010 IF b>3 THEN PRINT "Crash!!"
: LET l=l-10
5020 RETURN
```

## Vegas Breaker

Chuck Hopper's program — VEGAS BREAKER — is a variation on the old FRUIT MACHINE favourite, which costs you an inflationary $1.50 a spin. From time to time the HOLD option will come up. You can hold all four reels if you like. When HOLD comes up, you just enter each number you wish to hold, pressing ENTER after each one. When you have held enough, or if you don't want to hold any, enter 5, then press ENTER which gets you back to the next roll. Note line 40 (POKE 23692, −1) which keeps the screen scrolling, without you having to respond to a 'scroll?' query. The use of this POKE is discussed elsewhere in this issue, in Tim Hartnell's article on using Spectrum colour.

```
THIS IS ROUND 14

YOU HAVE $33

PRESS ANY KEY TO ROLL

   █**█ █$$█ ███ █**█

        YOU NOW HAVE $31.5
```

```
  10 REM VEGAS BREAKER
  20 POKE 23609,100
  30 GO SUB 9000
  40 POKE 23692,-1
  50 PAPER 0: CLS : BORDER 0: IN
K 7
  60 PRINT ''' PAPER 2;TAB 2;"TH
IS IS ROUND ";ROUND''TAB 2;"YOU
HAVE $";MONEY''TAB 2;"PRESS ANY
KEY TO ROLL"
  70 IF INKEY$<>"" THEN GO TO 70
  80 IF INKEY$="" THEN GO TO 80
  85 POKE 23692,-1
  90 FOR G=1 TO 50: BORDER RND*7
: BEEP .01,50-G: NEXT G: BORDER
0
 100 FOR J=1 TO 4
 110 IF M(J)=J THEN GO TO 150
 120 LET A(J)=INT (RND*4)+1
 140 BEEP .1,50/J
 150 NEXT J
 155 LET ROUND=ROUND+1
 160 GO SUB 5000
 165 GO SUB 4000
 170 IF RND>.7 THEN GO SUB 6000
 175 FOR T=1 TO 40: PRINT AT 1,2
5; INK RND*7;"█ █";AT 1,25; INK
RND*7;" █ ": NEXT T
 177 FOR T=1 TO 25: PRINT : NEXT
T
 180 IF MONEY>0 THEN GO TO 60
 190 PRINT '''TAB 5;"YOU SURVIVE
D ";ROUND;" ROUNDS"
 195 BORDER RND*7
 200 PRINT "BUT NOW YOU ARE BROK
E AND THE"
 205 BORDER RND*7
 210 PRINT "C A S I N O   I S   C
L O S E D!"
 215 BORDER RND*7
 220 POKE 23692,-1
 230 PAUSE 10
 240 GO TO 190
4000 REM ** MONEY **
4005 PRINT ''': POKE 23692,-1
4010 LET MONEY=MONEY-1.5
4020 IF A(1)=A(2) AND A(2)=A(3)
AND A(3)=A(4) THEN PRINT INK 6;"
£££££  JACKPOT!!!!  £££££££££":
 BEEP 2,10: PRINT ''"YOU WIN $10
!!": LET MONEY=MONEY+10: GO TO 4
100
4030 IF (A(1)=A(2) AND (A(3)=A(2
) OR A(4)=A(2))) OR A(1)=A(2) AN
D A(2)=A(4) OR A(2)=A(3) AND A(3
)=A(4) THEN PRINT INK 6; PAPER 2
;"$$$$$$$$ THREE OF A KIND! $$$$
$": BEEP 2,20: PRINT ''"YOU WIN
$5": LET MONEY=MONEY+5: GO TO 41
00
4040 IF A(3)=A(2) AND A(3)=A(4)
THEN PRINT INK 6; PAPER 2;"$$$$$
$$$ TRIO $$ TRIO $$ $$$$$": BEEP
 2,40: PRINT ''"YOU WIN $7.50":
LET MONEY=MONEY+7.5: GO TO 4100
4050 IF A(1)+A(2)+A(3)+A(4)=10 T
HEN PRINT PAPER 2;">>>>>>>>>"; PA
PER 0;" SMASHEROO!!": BEEP 2,-30
: PRINT ''"YOU WIN $7.50!!": LET
 MONEY=MONEY+7.5
4100 FOR T=1 TO 20: BORDER RND*7
: BEEP .01,T: NEXT T: BORDER 0
4105 PRINT
4110 FOR T=1 TO 64: PRINT INK RN
D*7;"█";: NEXT T
4120 PRINT ''TAB 8;"YOU NOW HAVE
$";MONEY''
4130 FOR T=1 TO 64: PRINT INK RN
D*7;"█";: NEXT T
4140 PRINT
4150 POKE 23692,-1: PRINT : PRIN
T
4160 DIM M(4)
4170 RETURN
5000 REM ** SPIN **
5010 FOR T=1 TO 50: BORDER RND*7
: BEEP .01,50/T/2: NEXT T: BORDE
5020 PRINT ''''TAB 4;
5030 FOR J=1 TO 4
5040 IF A(J)=1 THEN PRINT INK 2;
"█<>█ ";: BEEP .1,10
5050 IF A(J)=2 THEN PRINT INK 7;
"█$$█ ";: BEEP .1,20
5060 IF A(J)=3 THEN PRINT INK 4;
"█**█ ";: BEEP .1,30
5070 IF A(J)=4 THEN PRINT INK 5;
"████ ";: BEEP .1,40
5080 PAUSE 70
5090 NEXT J
5100 RETURN
6000 REM ** HOLD **
6010 DIM M(5)
6020 BEEP .5,1
6025 POKE 23692,-1
6030 PRINT '' INK 6;"ENTER ANY N
UMBER(S) YOU"
6040 PRINT ' INK 6;"WISH TO HOLD
, ENTER 5"
6050 PRINT ' INK 6;"WHEN YOU HAV
E FINISHED"
6060 INPUT Q
6070 IF Q<>5 THEN PRINT INK 2;Q
6080 LET M(Q)=Q
6090 IF Q<>5 THEN GO TO 6060
6100 RETURN
9000 REM ** ASSIGN VARIABLES **
9010 DIM A(5): DIM M(5)
9020 LET MONEY=7.5
9030 LET ROUND=1
9040 FOR T=1 TO 20
9050 BEEP .2,2*T
9060 NEXT T
9070 BORDER 7: PAPER 7: CLS
9080 BORDER 0: PAPER 0: CLS
9090 RETURN
```



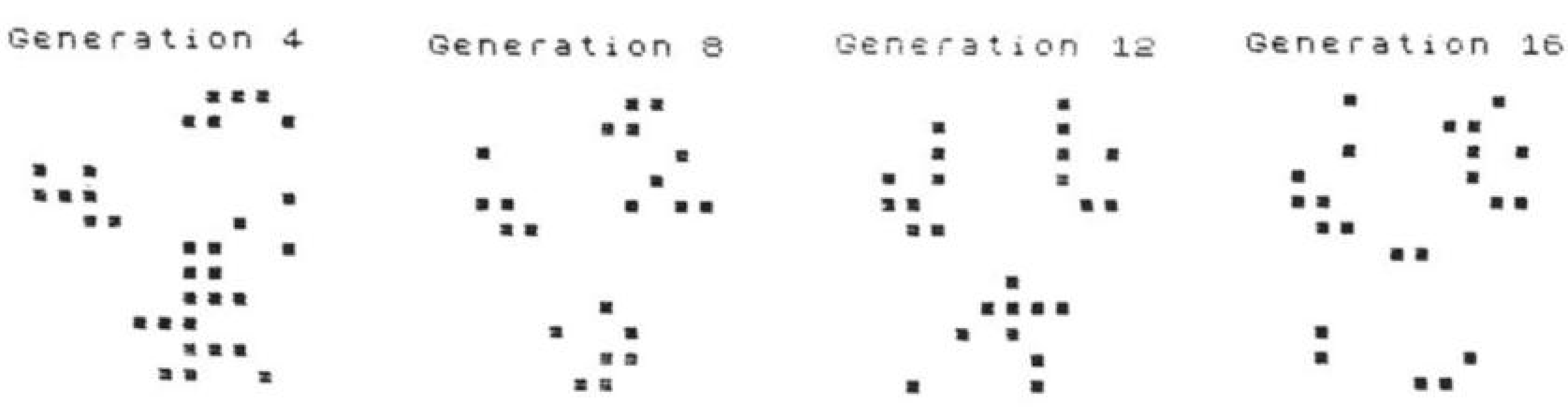



## Living colourfully

Anne Marshall has turned her inventive fingers to programming this variation of John Conway's game of LIFE. It makes good use of the colour available on Spectrum, and shows a novel approach to the program. We'll be discussing the game of LIFE in detail in the next issue of *ZX Computing*, and telling you how you can write a program to play it from scratch — working it out from the primary algorithm. But for now, save all that thinking, and give Anne Marshall's program a whirl.

*LIFE in progress.*

```
   5 REM LIFE - © ANNE MARSHALL
  10 DIM A(145): DIM L(145): DIM
 E(8)
  15 LET G=0
  20 FOR T=1 TO 8
  25 READ Z: LET E(T)=Z: NEXT T
  30 LET C=CODE "■": LET Z=128
  35 BORDER 1: PAPER 0: CLS
  40 FOR B=1 TO 12
  50 FOR D=1 TO 12
  60 LET A(B+10*D)=Z
  70 IF RND>.45 THEN LET A(B+10*
D)=C
  80 LET L(B+10*D)=A(B+10*D)
  90 NEXT D: NEXT B
 100 LET G=G+1
 120 FOR U=1 TO 12
 130 FOR B=1 TO 12
 140 LET F=U+10*B
 180 IF G=1 THEN GO TO 250
 190 LET H=0
 200 FOR T=1 TO 8
 210 IF A(F+E(T)+1)=C THEN LET H
=H+1
 220 NEXT T
 230 IF A(F)=C AND H<>3 AND H<>2
 THEN LET L(F)=Z
 235 IF A(F)=Z AND H=3 THEN LET
L(F)=C
 240 NEXT B: BORDER RND*7: NEXT
U
 245 BORDER 1
 250 FOR M=11 TO 144: LET A(M)=L
(M): BEEP .005,M/3: NEXT M
 255 PRINT AT 5,0;
 260 FOR U=1 TO 12: PRINT TAB 4;
 270 FOR B=1 TO 12: LET F=U+10*B
 280 PRINT INK 6;CHR$ A(F);" ";:
NEXT B: PRINT : NEXT U
 285 PRINT AT 3,10; PAPER 2; INK
 6;"Generation ";G: BEEP .3,50
 290 GO TO 100
 300 DATA 11,10,9,1,-1,-9,-10,-1
1
```






94

# Magical mischief

**Kar Wing Wong from Canada has provided us with a couple of brain-stretching programs for the 1K ZX80 — MAGIC SQUARE and ZX REVERSE. Both programs can easily be converted to run on the ZX81 (where more than 1K will be required) or the ZX Spectrum.**



When the MAGIC SQUARE program is first run, you'll see a three by three grid, partially filled with black blocks. The object of the game is to create a magic square by putting numbers in the place of the black blocks. The arrangement of the blocks can be changed by entering a number, from one to nine, with each number representing a position on the grid as follows:

1 2 3
4 5 6
7 8 9

When a number is entered, certain blocks will be reversed. A magic square is formed when the whole grid is black, except for the central square.

Here is the code to help you crack the Magic Square:
(1) When a number belonging to the corner (1,3,7 or 9) is entered, the colour of that quarter of the whole grid will be reversed. For example, when one is entered, the colour of the numbers 1, 2, 4 and 5 will be reversed.
(2) When a number corresponding to the middle of one of the four sides is entered (ie. 2, 4, 6 or 8), the colour of that whole side would be reversed. For example, when a four is entered, numbers 1, 4 and 7 would be reversed.

(3) When the centre one is entered (ie. number 5, numbers 2, 3, 4, 5, 6 and 8 would be reversed.

The magic square is made when the colour of numbers 1, 2, 3, 4, 6, 7, 8 and 9 are black, and the colour of number 5 is white. The computer will tell you the number of moves you made in order to finish the square, and it will only accept a number in the range from one to nine. However, the program will stop when you enter a letter, if you feel living giving up.

```
  5 CLS
 10 DIM A(9)
 20 LET B=0
 30 FOR A=1 TO 9
 40 LET K=RND(2)
 50 IF K=1 THEN LET A(A)= 128
 60 IF K=2 THEN LET A(A)=0
 70 LET B=B+K
 80 NEXT A
 90 IF B=8 AND A(5)=0 THEN RUN
100 FOR N=0 TO 2000
110 LET C=0
120 PRINT"MAGIC SQUARE"
130 PRINT"------------"
140 PRINT
```

```
150 LET D=0
160 PRINT"(AAAAAAA)"
170 FOR A=1 TO 9
180 PRINT"(A)":CHRS$(A(A));
190 IF (A/3)*3=A THEN PRINT"(A)"
200 IF (A/3)*3=A THEN PRINT"(AAAAAAA)"
210 LET D=D+A(A)
220 IF D=1024 AND A(5)=0 THEN GOTO 470
230 NEXT A
240 IF C=1 THEN PRINT B;" IS NOT BETWEEN 1 AND 9,
    DUMMY"
250 PRINT
260 PRINT"ENTER A NUMBER (1 TO 9)"
270 INPUT B
280 CLS
290 IF B < OR B > 9 THEN GOTO 540
300 LET A(B)=ABS(A(B)-128)
310 LET K=1
320 IF B=3 OR B=4 THEN LET K=B-1
330 IF B=6 OR B=7 THEN LET K=9-B
340 IF NOT(B/2)*2=B AND NOT B=5 THEN GOTO 410
350 IF B=5 THEN GOTO 370
360 GOTO 430
370 FOR K=1 TO 4
380 LET A(2*K)=ABS(A(2*K)-128)
390 NEXT K
400 NEXT N
410 LET A(5)=ABS(A(5)-128)
420 LET B=(5-B)/2+B
430 LET A(B+K)=ABS(A(B+K)-128)
440 LET A(B-K)=ABS(A(B-K)-128)
450 NEXT N
460 STOP
470 PRINT"YOU DID IT IN ";N;" MOVES"
480 CLEAR
490 PRINT"TYPE Y TO PLAY AGAIN"
500 INPUT A$
510 IF A$="Y" THEN RUN
520 CLEAR
530 LIST
540 LET C=1
550 GOTO 120
```

## ZX Reverse

When this game runs, you'll see a random sequence of numbers, from one to nine. The object of the game is to get the numbers back in order again in as few moves as possible.

Here's how you do it. If the numbers were arranged 286491537, and you entered 5, then the sequence will become 946821537, that is, the first five numbers would change places. The computer keeps track of the number of moves you've made and will know when you've got the numbers back in order.

```
5 LET C=0
10 DIM A(9)
20 LET A(1)=RND(9)
30 FOR A=2 TO 9
40 LET A(A)=RND(9)
50 FOR B=1 TO A-1
60 IF A(A)=A(B) THEN GOTO 40
70 NEXT B
80 NEXT A
90 FOR D=0 TO 2000
100 CLS
110 PRINT,"ZX REVERSE"
120 PRINT,"----------"
130 PRINT
140 PRINT"  ";
150 FOR B=1 TO 9
160 PRINT A(B);"  ";
170 NEXT B
180 PRINT
190 PRINT
200 FOR B=1 TO 9
210 IF NOT A(B)=B THEN GOTO 240
220 NEXT B
230 GOTO 470
240 IF C=0 THEN GOTO 280
250 PRINT
260 PRINT" PLEASE INPUT AS INSTRUCTED"
270 PRINT" I AM JUST A DUMB COMPUTER"
275 PRINT" YOU KNOW"
280 PRINT
290 PRINT" ENTER A NUMBER(2 TO 9)"
300 PRINT" OR TYPE 1 TO STOP"
310 INPUT A$
320 IF A$="1" THEN GOTO 520
330 FOR A=30 TO 37
340 IF A$=CHR$(A) THEN GOTO 370
350 NEXT A
360 GOTO 540
370 LET C=0
380 LET A=A-28
390 FOR B=1 TO A/2
400 LET E=A(A)+A(B)
410 LET A(A)=E-A(A)
420 LET A(B)=E-A(A)
430 LET A=A-1
440 NEXT B
450 NEXT D
460 STOP
470 PRINT" YOU DID IT IN ":D" MOVES"
480 PRINT
490 PRINT" TYPE Y TO PLAY AGAIN"
500 INPUT A$
510 IF A$="Y" THEN RUN
520 CLEAR
530 LIST
540 LET C=1
550 GOTO 100
```



# MICHAEL ORWIN'S ZX81 CASSETTES

**THE BEST SOFTWARE (BY VARIOUS AUTHORS) AT LOW PRICES**

## QUOTES

"Michael Orwin's £5 Cassette Two is very good value. It contains 10 stolid well designed games which work, offer plenty of variety and choice, and are fun."
from the ZX Software review in
*Your Computer*, May '82 issue

"I had your Invaders/React cassette . . . I was delighted with this first cassette."
P. Rubython, London NW10

"I have been intending to write to you for some days to say how much I enjoy the games on 'Cassette One' which you supplied me with earlier this month."
E. H., London SW4

". . . I previously bought your Cassette One and consider it to be good value for money!"
Richard Ross-Langley
Managing Director
Mine of Information Ltd.

## CASSETTE 1
(eleven 1k programs)

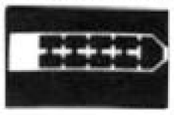
machine code:
React, Invaders, Phantom aliens, Maze of death, Planet lander, Bouncing letters, Bug splat.
Basic:
I Ching, Mastermind, Robots, Basic Hangman PLUS Large screen versions of Invaders and Maze of Death, ready for when you get 16k.
**Cassette One costs £3.80**

## CASSETTE 2
Ten games in Basic for 16k ZX81

Cassette Two contains Reversi, Awari, Laser Bases, Word Mastermind, Rectangles, Crash, Roulette, Pontoon, Penny Shoot and Gun Command.
**Cassette Two costs £5**

## CASSETTE 3
8 programs for 16k ZX81

**STARSHIP TROJAN**



Repair your Starship before disaster strikes. Hazards include asphyxiation, radiation, escaped biological specimens and plunging into a Supernova.

**STARTREK** This version of the well known space adventure game features variable Klingon mobility, and graphic photon torpedo tracking.

**PRINCESS OF KRAAL** An adventure game.

**BATTLE** Strategy game for 1 to 4 players.

**KALABRIASZ** World's silliest card game, full of pointless complicated rules.

**CUBE** Rubik Cube simulator, with lots of functions including 'Backstep'.

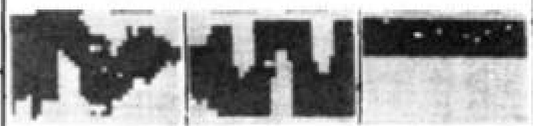
**SECRET MESSAGES** This message coding program is very txlp qexi jf.

**MARTIAN CRICKET** A simple but addictive game (totally unlike English cricket) in machine code. The speed is variable, and its top speed is very fast.

**Cassette 3 costs £5**

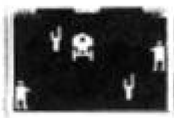
## CASSETTE 4
8 games for 16k

**ZX-SCRAMBLE** (machine code)

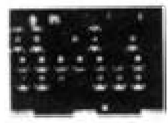


Bomb and shoot your way through the fortified caves.

**GUNFIGHT** (machine code)



**INVADERS** (machine code)



**GALAXY INVADERS** (machine code)
Fleets of swooping and diving alien craft.

**SNAKEBITE** (machine code)
Eat the snake before it eats you. Variable speed (very fast at top speed).

**LIFE** (machine code)
A ZX81 version of the well known game.

**3D TIC-TAC-TOE** (Basic)
Played on a 4x4x4 board, this is a game for the brain. It is very hard to beat the computer at it.

7 of the 8 games are in machine code, because this is much faster than Basic. (Some of these games were previously available from J. Steadman).
**Cassette 4 costs £5**

**FUNGALOIDS** (Machine code)

Recorded on quality cassettes, sent by first class post, from:
**Michael Orwin, 26 Brownlow Road, Willesden, London NW10 9QL** (mail order only please)

---

EAST LONDON ROBOTICS

The specialists in internal plug-in memories for ZX computers announce:

# 80K SPECTRUM

now attainable for the price of a 48K model!! with our SP80 low-power Sinclair look-alike.

Functionally identical to the Sinclair 32K internal plug-in expansion but with double the capacity (64K) the SP80 plugs into the sockets provided on the 16K SPECTRUM by Sinclair for his 32K expansion board. Instructions to our usual high standard makes fitting very simple indeed.
The SP80 in no way interferes with Sinclair add-ons — ZX Printer, RS232 interface, Microdrive . . .

New low prices on our highly successful internal memories for ZX81
To: East London Robotics, 'Finlandia House', 14 Darwell Close, LONDON E6 4BT.

| Item | Item Price | Quantity | Total |
|---|---|---|---|
| CHIPSWITCH kit doubles your ZX81 memory to 2K. (this kit requires soldering) | 4.70 | | |
| INCREMENTAL internal 2K plug-in memory extension for ZX81 expandable to 16K. | 17.75 | | |
| Additional 2K chips for above (HM6116P-3). | 4.50 | | |
| MAXIMEM 64K internal plug-in memory for ZX81. | 49.95 | | |
| SP80 64K internal plug-in memory extension for ZX SPECTRUM giving 80K of user RAM. | 50.00 | | |
| SP80 Kit version with full instructions. | 44.00 | | |
| SP80 fitting service (price includes excess P&P). | 9.00 | | |
| Postage and Packing | | | 0.45 |
| | | TOTAL £ | |

Telephone enquiries on 01 471 3308.
All prices already include VAT.
Please tick if you require a VAT receipt.
Refunds less £1.50 handling on all items returned within 14 days of receipt. Send stamped addressed envelope plus additional 12½p stamp for catalogue.

Cheque/Postal Order payable to: EAST LONDON ROBOTICS £ ____________

Name: Mr/Mrs/Miss . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Adrress: . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

STOP PRESS!!! Transfer all your ZX81 BASIC and Machine code programs and data onto your SPECTRUM in minutes with our new SLOWLOADER available soon!!

---

Great games packs for 16K ZX81

**PACK 1**

# ASTRO-INVADERS

Just look at these features . . .
. . . then look at the price!

*Superior machine code programming
*Rapid-firing with explosive on-screen kill effect
*High-scoring saucers *54 aliens *accelerating attack
*Destructable defence shields *On-screen kill count
*High-score update
*Fast action space graphics
— a new demension in ZX81 value

Astro-Invaders is yours on cassette for **ONLY £3.65** with **FOUR BONUS GAMES:**
**ARCADE GRAND-PRIX** — drive four levels of machine code skill
**PENALTY** — defend your goal against the sharp-shooting ZX81
**GOLF** — judge shot-strength, angles, bunkers . . .
plus machine code insect fun with **SWAT**

**PACK 2**

# PLANET DEFENDER

— blast aliens in planet orbit . . .

*ultra-dynamic machine code action *hostile alien-waves
*fast, responsive controls: ship up/down, thrust, laser-bolt, and smart bomb *comprehensive scoring
*explosive graphics . . . . . . explosive price!

Planet Defender comes on cassette for **ONLY £3.65** with machine code **STORM-FIGHTERS** — combat 5 fleets of swooping aliens as you blast across space, **BREAKOUT** (machine code) — race against the clock, plus **GRAPHIC HANGMAN.**

**16K ZX81 cassette packs 1 and 2 are £3.65 each (post free in UK) from:**

**JOHN PRINCE**
29 Brook Avenue, Levenshulme,
Manchester, M.19

# User-definable graphics

**Thirteen-year-old Chris Callender from Cove, Helensburgh, has devised a great program to allow you to define your own characters for dumping to the ZX printer.**

This program will work with a ZX81 or an 8K ROM ZX80. It needs a printer, and 16K RAM. The first thing you must do before typing in the program, or LOADing it from cassette, is to type:

POKE 16389, 124

NEW

This will alter RAMTOP to make space for the machine code subroutine. Next, type in the program as listed, and press RUN.

Now, to design, say, an arrow like the one in figure one, type:

".";NEWLINE
"...";NEWLINE
"...";NEWLINE
"...";NEWLINE
".";NEWLINE
".";NEWLINE
".";NEWLINE
".";NEWLINE

The program will then be ready for the next character on that line. If you are finished, type: "PLOT". There will be a delay of about 10 seconds and then there should be a buzz from the printer and there will be your character.

To design a character, draw an eight by eight grid and make up your character by filling in the segments of the grid. Then type in your character line by line. Everytime you come up against a blob that should be filled in, enter a fullstop. You make spaces by, obviously enough, typing a space.

## How it works:

The program works by using a machine code routine copied from the ROM at address 2161. This is copied above RAMTOP by lines 5-9. Then the user enters a line of his character as B$. Lines 50-120 convert this line to a byte of information in array A$. If the user types PLOT, the program goes to line 9988 which is the start of the print routine.

| LINE NUMBER: | EFFECT: |
|---|---|
| 1 | Checks if memory above RAMTOP has been reversed |
| 2 | Gives error code if not |
| 5-9 | Copy and adapt print routine in ROM and set up above RAMTOP |
| 12 | Sets up array A$ to store characters |
| 30 | Displays current line |
| 40 | Input line of the character |
| 50-120 | Sets C to the right code for array A$ |
| 130 | Sets array A$ to C |
| 9988-9999 | Sinclair's high resolution printing program |

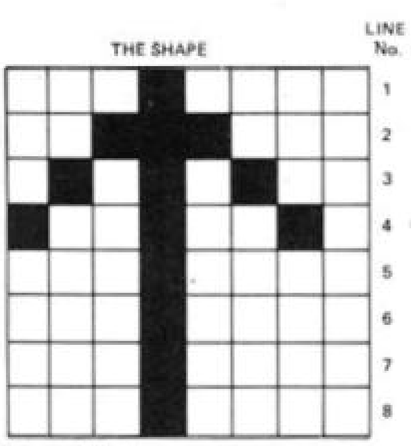
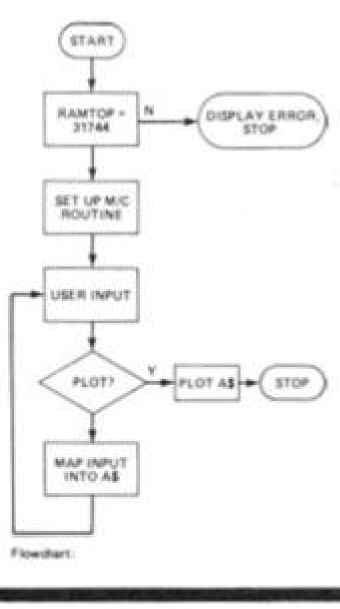
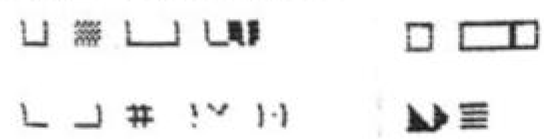


Figure one: Drawing an arrow



Flowchart

```
   1 IF PEEK 16388+256*PEEK 1638
9=31744 THEN GOTO 5
   2 POKE 16384,21
   5 FOR I=0 TO 112
   6 POKE 31744+I,PEEK (2161+I)
   7 NEXT I
   8 POKE 31800,63
   9 POKE 31857,201
  12 DIM A$(32,9)
  15 FOR N=1 TO 32
  20 FOR A=1 TO 8
  25 LET C=0
  30 PRINT AT 0,0;"LINE ";A
  40 INPUT B$
  41 PRINT AT A,0;B$
  42 IF B$="PLOT" THEN GOTO 9988
  43 IF LEN B$<8 THEN LET B$=B$+
"        "
  44 IF B$( TO 4)="EXIT" THEN PO
KE 16384,13
  50 IF B$(1)="." THEN LET C=C+2
**7
  60 IF B$(2)="." THEN LET C=C+2
**6
  70 IF B$(3)="." THEN LET C=C+2
**5
  80 IF B$(4)="." THEN LET C=C+2
**4
  90 IF B$(5)="." THEN LET C=C+2
**3
 100 IF B$(6)="." THEN LET C=C+2
**2
 110 IF B$(7)="." THEN LET C=C+2
 120 IF B$(8)="." THEN LET C=C+1
 130 LET A$(N,A+1)=CHR$ C
 135 PRINT AT 21,0;C
 140 NEXT A
 145 CLS
 150 NEXT N
9988 REM LPRINT A$ 8 LINES AT A
TIME
9990 FOR J=1 TO 32
9991 FOR K=1 TO 8
9992 POKE 32255+K+8*(J-1),CODE A
$(J,K+1)
9993 NEXT K
9994 NEXT J
9995 FOR H=0 TO 31
9996 POKE 16444+H,H
9997 NEXT H
9998 LET HPRINT=USR 31744
9999 POKE 16384,26
```



```
LINE 1
LINE 2
LINE 3
LINE 4
LINE 5
LINE 6
LINE 7
LINE 8
LINE 1
ETC. (PROGRAMING OTHER ARROWS
PLOT
```

↑ → ← ↓

# HINTS 'N' TIPS TO IMPROVE YOUR PROGRAMS

**Experienced programmers develop many useful techniques, but they rarely get the chance to pass the results of that experience on. Here, Dilwyn Jones from North Wales shares with you a host of ideas to help you polish up your programming skills.**

In this article, I've brought together a number of things I've learned while working with my ZX81. Many of the hints will apply to working with the ZX Spectrum.

The first thing I'd like to discuss is the fact that, when using the 1K ZX81, the amount of screen memory used is a vital consideration. Because the display lines are expanded in memory only when something is printed on screen, it follows that the more you print, the more memory that you use up. Also printing towards the right hand side of the screen will eat up memory because the ZX81 has to fill out the line before what you've printed with spaces, so try to print on the left hand side wherever you can. Also, if there are a lot of redundant print statements on screen then use CLS often to get rid of them.

When editing a listing, you sometimes have to move the cursor up or down a long way to get to the line you want to edit. Suppose you want to edit line N. The instruction LIST N will put the cursor at the top of the screen in line N, so you can now edit the line quickly.

If you have problems editing when you are nearly out of memory, then try this method. Use LIST N as above to place the line required at the top of the screen, then press CLEAR (or CLS if you want to preserve variables in memory) to make space in memory. It does not matter that you can't see the listing. Now press EDIT (shift 1) and the line appears ready for editing.

If you have LOAD problems then try the following tips:

(1) Disconnect the lead not in use from both the ZX81 and the cassette recorder.

(2) Try operating the cassette recorder from batteries.

(3) Try moving the ZX81 and the cassette recorder further apart, as well as the TV if you can.

(4) Change the volume setting on the cassette recorder since some cassettes may have a higher output than others. Try changing the tone control settings, in particular turn up the treble or turn down the bass.

(5) Make sure your leads have not broken or cracked, or a solder joint could have come loose.

(6) Do not try loading a 1K program SAVEd using a RAM Pack because although the program itself might easily fit into 1K, the display file will be at its full size and so there will be no room for everything in the ZX81. The answer to this is to ensure that the display file is contracted to minimum size before saving (if you have more than 3¼K of memory) by setting RAMTOP to, say, 17408 (1K of RAM) followed by clearing the screen before saving. If you've got a program saved using a RAM pack and which won't load then beg, steal, borrow or even buy a RAM pack and go through the motions described above and re-save the program to make it suitable for loading into 1K in future.

(7) This sounds silly, but make sure your plugs are in the correct hole! You may find it useful to stick labels on top of the ZX81 above the sockets to tell you which one is which so that you don't have to peer round the side to look every time.

To avoid flicker when using PAUSE, replace with a FOR/NEXT loop, eg. 10 PAUSE 500 could be replaced by

```
10 FOR F = 1 TO 600
20 NEXT F
```

A loop of 60 corresponds approximately to one second in practice as does PAUSE 50.

If you can, use the same variable for as many purposes as you can, especially when you use FOR/NEXT loops. Don't use another letter as the name for a second FOR/NEXT loop if you've already finished with a previous one as this would be wasteful of memory.

You can use this expression to PRINT any of the graphics characters or their inverses at random:

PRINT CHR$(RND x 10 + (128 AND RND < .5))

GOSUB or GOTO destinations don't have to be absolutely correct. If the program doesn't find the line number you've specified, then it will go to the next highest number.

If you want a time limit on user responses without involving the frame counter, use this method. Suppose the user had two seconds to decide whether or not to have another game. If he/she was too slow deciding then the program stopped. For the purpose of this routine suppose the user had to press R for a re-run:-

```
10 FOR F = 1 TO 55
20 LET A$ = INKEY$
30 IF A$ = "R" THEN
   GOTO 60
40 NEXT F
50 STOP
60 PRINT "RE-RUN"
70 RUN
```

In view of a previous paragraph you may have expected to have a loop of 120 in line 10, but having anything inside the FOR/NEXT loop slows it down and 55 is adequate in this case.

The frame counter is a bit more difficult to manipulate on the ZX81 than its predecessor. To obtain the same values we need to do some careful conversions because the frame counter counts backwards on the ZX81, starts off from 65536 and counts down to 32768. This is because bit 15 is normally set to 1. To use the

frame counter as a timer use this routine to first set the timer:

```
POKE 16437,255
POKE 16436,255
```

and to read its value at any time use

```
LET T = (65536 - PEEK
16436 - 256 x PEEK
16437)/50
```

which will give you a fairly accurate readout in seconds if you PRINT T. You may have to subtract a fraction of a second to allow for the time taken to work out the expression. Remember that PAUSE uses the frame counter so it cannot be used for timing if you are using PAUSE in your program.

If you wanted to PRINT a certain amount of characters, for example if you wanted to draw a line of "–" characters for underlining, then here are 2 methods. Obviously, different headings will be of different lengths, so you need to know how many characters to PRINT. If you're printing a string, eg. A$ you use the function LEN to tell you the length of A$, hence this is the amount of characters to PRINT.

(1)

```
10 FOR A = 1 TO LEN A$
20 PRINT "–";
30 NEXT A
40 PRINT
```

Line 40 moves the PRINT position to the next line ready to continue. Omit it if you do not need it. The next method is a lot faster and uses only one program line.




(2)

```
PRINT "------------------------"
(TO LEN A$)
```

the only disadvantage is that you need to specify how many characters are required in quotes even though they may never be printed. That is, you need to know the longest that A$ can possibly be so that you can put that many characters in the string constant in quotes after PRINT.

TAB reduces a number modulo 32, meaning that the argument of (number after) TAB can be larger than 31; it will be reduced to a number in the range 0 to 31 and the PRINT position moves on the same line unless this would involve backspacing in which case it moves onto the next line. What this modulo business means is that the argument of TAB is divided by 32 (the number of columns per line on a screen) and the remainder taken. You may be able to take advantage of this when the PRINT spacing is determined by calculation since you do not have to ensure that the number falls in the range 0 to 31.

Try applying VAL to an expression like "ATN 1 x 4" : it works, and this is often quite an useful facility. Also you can have the name of a numeric variable in quotes and provided it has previously been defined or assigned, it will be successfully evaluated. In fact VAL can be applied to all sorts of numeric expressions, and is sometimes useful to replace the DEF FN function found in other BASICs. It may also be useful if you wish to generate random numbers several times in a program. At the start of the program have a statement like A$ = "RND x 6" and every time you wanted a random number you would type LET R = VAL A$

In a FOR/NEXT loop, STEP does not have to be a whole number; it may be a fraction, decimal, the result of a calculation and does not have to hit the limit value of the loop exactly. It carries on looping as long as it is less than or equal to the limit. You cannot easily change the value of STEP during the course of a loop.

If the limit value has already been exceeded then the loop will be totally bypassed, eg.

```
10 FOR F = 1 TO 0
20 PRINT "X"
30 NEXT F
```

You may be able to use this idea to prevent loops being executed if certain conditions exist, eg. if you didn't want a black line to be drawn if X was equal to 6:

```
1000 FOR F = (X = 6) x
     33 TO 31
1010 PRINT CHR$ 128;
1020 NEXT F
```

The test for whether the limit value has been exceeded is made at the line containing the FOR statement. An interesting experiment is to try a STEP value of 0. The control variable is never incremented and so the loop never ends! You can jump out of FOR/NEXT loops without any problems, but you cannot jump into a loop unless the control variable has already been set up (effectively if you've used that loop before). In a FOR/NEXT loop jumps from NEXT to the line following the FOR statement. Some versions of BASIC allow you to omit the variable after NEXT and the most recent control variable is then incremented; you must specify the control variable on the ZX81.

Because you can use FAST and SLOW as program statements, you can switch from one to the other in programs that require that patterns are displayed only for a certain length of time, or you can switch into FAST to initialise a program or to POKE machine code into memory for example.

Some programs require that the screen be cleared occasionally to prevent a screen memory overflow when the PRINT position gets down to the bottom of the screen. Here is one way to do this:-

```
IF PEEK 16442 < 4
THEN CLS
```

16442 is the system variable




containing the line number of the PRINT position. It starts off at 24 for the top line, down to 3 for the lowest line available to the programmer and 2 and 1 for the two lines at the bottom of the screen used for INPUT etc. I have used 4, but you could substitute another number if you like.

Normally you can only PRINT on the top 22 lines of the screen display (lines 0 to 21). Any attempt to use the bottom two lines with PRINT is normally rewarded by an error report 5. You can gain access to these lines by two methods. The simplest is to POKE directly into memory at the location of the bottom two lines of the screen. If you have more than 3¼K of memory plugged in (eg. if you have a 16K RAM Pack) so that if the display is at full size then line 22 starts at (PEEK 16396+256 x PEEK 16397+727), ends at (PEEK 16396+256 x PEEK 16397+758). Line 23 consequently starts at (PEEK 16396+256 x PEEK 16397+760) and ends at (PEEK 16396+256 x PEEK 16397+791). These addresses will be different if th display file size is altered, as might happen if SCROLL was used. The second method uses PRINT AT and the system variable DF - SZ at address 16418. The number in 16418 says how many lines in the bottom of the screen are not available to the user — normally two. So if we change this number to 0, we have access to all 24 lines of the screen display and we can use PRINT AT 23,X or PRINT AT 22,X. However, this method comes unstuck when the computer tries to use the bottom of the screen for error reports, INPUTs, or even SCROLL. You can get a very nasty system crash and lose your program if you're unlucky (no lasting damage will be done, but you may have to switch off for a few seconds). The statement POKE 16418,0 must be entered as a line in a program. It does not work if entered as a direct command without a line number because the computer will reset it automatically when the screen is cleared, or a program is RUN. If you wish to use INPUT during the course of a program is RUN. If you wish to use INPUT during the course of a program then you should POKE 16418,2 to restore the bottom of the screen to normal before attempting to use INPUT, which will of course erase characters PRINTed on line 22 and 23! Incidentally, be careful if you're using an unexpanded machine — the display file behaves in a strange way and makes use of 16418 so try not to upset it too much!

To place any particular line number you require at the top of automatic listings, you must first move the cursor to a line number greater than the one you want at the top.

Then enter:

POKE 16419,NUMBER – INT (NUMBER/256) x 256
POKE 16420,INT(NUMBER/256)

Now when you press NEWLINE the automatic listing will begin where you specified (NUMBER is the line you want at the top of the screen). When entering lines when the cursor is at the bottom of the screen, the ZX81 will usually compile the listing 2 or 3 times to get the new line onto the screen listing at the bottom. This is annoying, not to mention time-consuming. You can circumvent this like this: Type in any line number *higher* than any shown on screen and which does not exist in the listing (I always use 9999). The listing will change. If you now continue entering lines where you were originally, they appear near the top of the screen and the listing is made properly, saving a lot of frustration.

# Specialist Books

**Choosing programs for microcomputers**
1980 J E Lane *£9.00*
A5 138pp P ISBN 0 85012 255 4
Looks at application packages for micros describing what they are, the benefits they offer and their use on microcomputers. Guidelines for obtaining packages and for identifying the best product are given.

**Elements of BASIC**
1979 R Lewis and B H Blakeley *£9.00*
A5 200pp P ISBN 0 85012 118 3
Introduces the BASIC language, covering the mathematical, non-numeric and data processing facilities. Generally machine independent with supplements to show the effect of a number of different implementations.

**Graphics on microcomputers**
1981 J E Lane *£4.00*
A5 44pp P ISBN 0 85012 333 X
Explores the type of graphics becoming increasingly available in low cost systems. Illustrates the facilities available and takes a closer look at graphics picture building techniques.

**Information handling by microcomputers**
1981 J E Lane *£4.00*
A5 60pp P ISBN 0 85012 334 8
Examines the field of information handling on microprocessors across the whole spectrum of micro applications. Aims to promote an awareness of current practices and trends.

NCC
The National Computing Centre

**Introducing computer programming**
1979 Reprint W G Collin *£11.50*
A5 364pp P ISBN 0 85012 210 4
A machine language independent textbook for the beginner, providing all the necessary basic information needed by someone starting on a computer programming career.

**Introducing data processing**
1980 NCC *£6.50*
A5 237pp P ISBN 0 85012 245 7
Covers the requirements of syllabi for introductory courses. Provides a comprehensive and accessible introduction to data processing. Assumes no previous knowledge of the subject.

**Introducing microprocessors**
1979 G L Simons *£9.00*
A5 177pp P ISBN 0 85012 209 0
Gives a profile of the microprocessor scene paying attention to typical application areas together with hardware and software information.

**Introducing word processing**
1981 G L Simons *£8.50*
A5 180pp P ISBN 0 85012 320 8
Describes the main characteristics of word processing and discusses its advantages over conventional typewriting. Communication, maintenance, security and costs are considered.

**Operating systems for microcomputers**
1981 J E Lane *£3.50*
A5 77pp P ISBN 0 85012 277 5
Establishes the requirements of operating systems for microcomputers in both commercial and industrial application areas and examines the facilities provided in a number of current products.

**Student notes on NCC DP documentation standards**
1978 NCC *£5.50*
A5 100pp P ISBN 0 85012 339 9
A subset of the full documentation standards for use by students on courses where NCC standards are part of the syllabus.

**The robots are coming**
1974 F H George & J D Humphries (eds) *£10.00*
A5 188pp P ISBN 0 85012 114 0
Gives a general background to current developments in artificial intelligence research and looks at where these developments could be leading.

**Using computers — a manager's guide**
1980 M Peltu *£7.50*
A5 180pp P ISBN 0 85012 241 4
Intended to help managers implement computer systems effectively in an organisation. Provides an introduction for user management covering the topics of planning and control plus human factors.

**Working with computers: a guide to jobs and careers**
1975 *£2.50*
A5 86pp P ISBN 0 85012 126 4
A general introduction to computing as a career for school leavers. Covers how a computer is used, what types of job exist and how to train for them.

NCC
The National Computing Centre

We are now able to offer, in addition to our usual selection of books on computers, a number of specialist titles from the National Computing Centre.

Rather than taking their entire list of some 110 titles, we have selected those most relevant to the microcomputer market and these are listed with their precis.

Ordering couldn't be simpler, just tick the boxes in the form below, enclose a cheque or postal order to the total amount (or make use of the Barclaycard and Access facility) and send it all off to:

**SPECIALIST BOOKS,**
**ZX COMPUTING**
**145 CHARING CROSS ROAD,**
**LONDON WC2H 0EE.**

If you are using your credit card to order please don't send it, just fill in the number and sign on the dotted line.
Please allow 28 days for delivery of your books.

☐ CHOOSING PROGRAMS FOR MICROCOMPUTERS *£9.00*
☐ ELEMENTS OF BASIC *£9.00*
☐ GRAPHICS ON MICROCOMPUTERS *£4.00*
☐ INFORMATION HANDLING BY MICROCOMPUTERS *£4.00*
☐ INTRODUCING COMPUTER PROGRAMMING *£11.50*
☐ INTRODUCING DATA PROCESSING *£6.50*
☐ INTRODUCING MICROPROCESSORS *£9.00*
☐ INTRODUCING WORD PROCESSING *£8.50*
☐ OPERATING SYSTEMS FOR MICROCOMPUTERS *£3.50*
☐ STUDENT NOTES ON NCC DP DOCUMENTATION STANDARDS *£5.50*
☐ THE ROBOTS ARE COMING *£10.00*
☐ USING COMPUTERS — MANAGER'S GUIDE *£7.50*
☐ WORKING WITH COMPUTERS: A GUIDE TO JOBS AND CAREERS *£2.50*

**Name:** ..........................................................

**Address:** ..........................................................

..........................................................

..........................................................

**Amount:** ...........

**Make cheques payable to ASP Ltd.**

We welcome Access

BARCLAYCARD VISA

I wish to pay by
BARCLAYCARD ☐ ACCESS ☐ tick
Card No.
☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

**Signature:**

# Laying it down, picking it up

**Brain-teasers have been sources of popular computer games from the days when the only computer game time was that stolen from companies which owned large mainframes. NIM, and variations on the 'he who picks up the last one, loses' theme, was one widely programmed game. We've got a version of it here, listed for the ZX81 and for the Spectrum. Board games also proved good sources of ideas for computer games. The popular FOUR IN A ROW comes to you now for the 16K ZX81.**

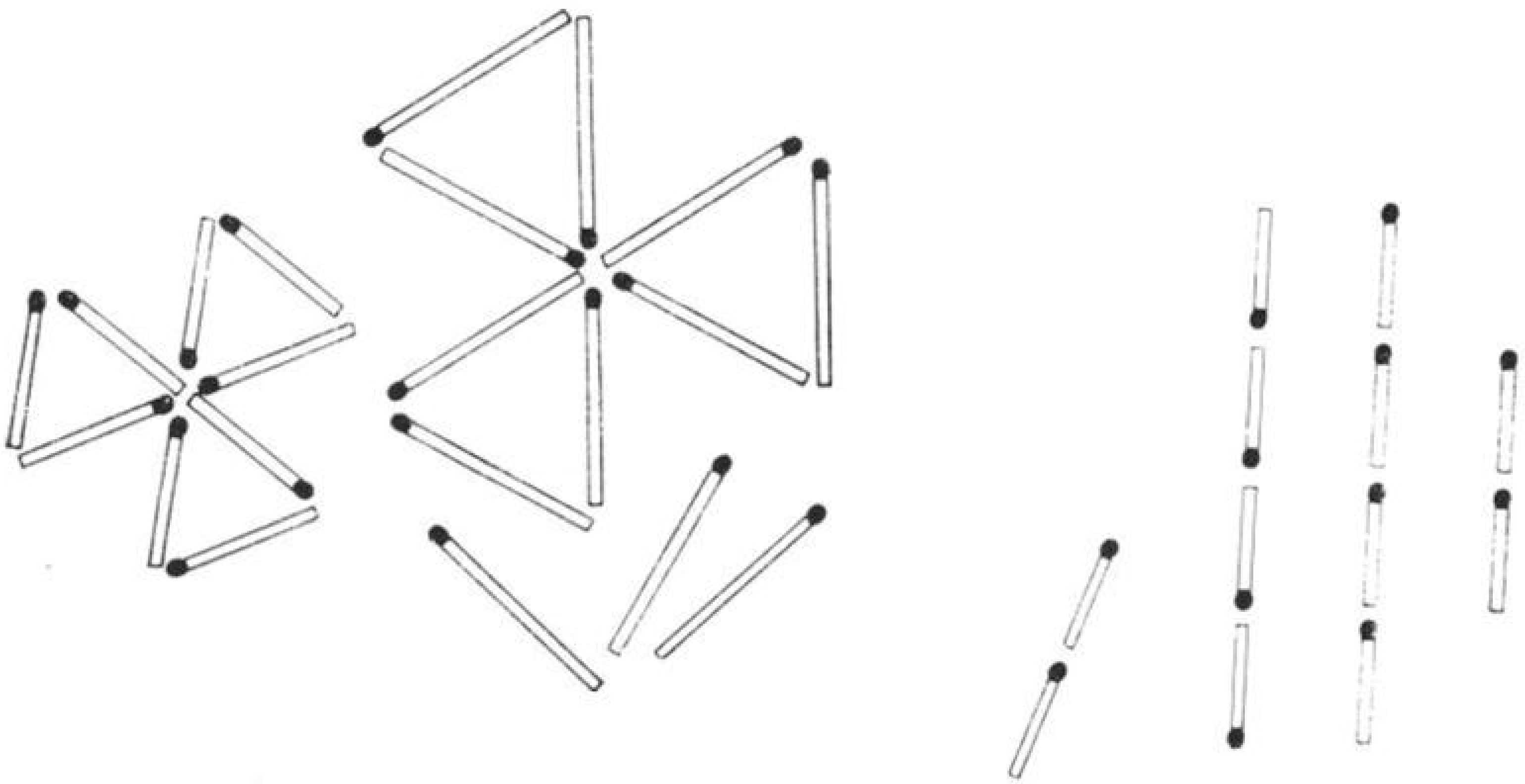
## Matchsticks

### For ZX81 or Spectrum, this variation on an old favourite will get you thinking.

The computer plays the human in a variation of the old 'player who picks up the last one loses' game. This game, Matchsticks, is based on one which was played in the film 'Last Year at Marienbad'. There are a certain number of matches at the start of the game, determined by the program, and there is a limit to how many you can pick up at a time. You and the computer take it in turns to take away as many matches as you choose, up to the maximum allowed.

Note that the Spectrum uses white text on a blue background with a blue border. This is one of the easiest to read PAPER/INK combinations, but feel free to change it to a combination of your choice.

The player who takes the last match loses in this game. The computer is not programmed to be infallible, so you have some chance of winning.



## The Spectrum Listing

```
   5 REM * MATCHSTICKS *
  10 REM WHITE TEXT ON BLUE
  15 PAPER 1: INK 7: BORDER 1: C
LS
  20 LET E=0: LET Z=16+INT (RND*
9)
  30 IF 2*(Z/2)=Z THEN LET Z=Z+1
  40 LET H=INT (RND*4)+2
  50 PRINT PAPER RND*5+2; INK 0;
AT 0,6;"MAXIMUM TO TAKE IS ";H
  60 IF E>0 THEN PRINT AT 7,2;"Y
OU TOOK ";E;TAB 20;"I TOOK ";Q
  70 FOR K=1 TO Z: BEEP .01,K
  80 PRINT INK RND*5+2;K;"█ ";
  90 IF RND>.85 THEN PRINT : PRI
NT
 100 NEXT K
 105 LET K=7: IF RND>.5 THEN LET
 K=4
 110 INPUT INK K;"HOW MANY WILL
YOU TAKE? ";E
 120 IF E>H OR E<1 THEN GO TO 11

 100 NEXT K
 105 LET K=7: IF RND>.5 THEN LET
 K=4
 110 INPUT INK K;"HOW MANY WILL
YOU TAKE? ";E
 120>IF E>H OR E<1 THEN GO TO 11
0
 130 CLS : LET Z=Z-E
 140 IF Z=0 THEN BORDER RND*7: P
RINT PAPER RND*5;AT 10,12;"I WIN
": BEEP .05,RND*30+30: GO TO 140
 150 LET Q=Z-1-INT ((Z-1)/(H+1))
*(H+1)+INT (RND*3)-1
 160 IF Q>Z OR Q<1 OR Q>H THEN G
O TO 150
 170 LET Z=Z-Q
 180 IF Z=0 THEN BORDER RND*7: P
RINT ; PAPER RND*6;AT 10,5;"I TO
OK ";Q;", SO YOU WIN!": BEEP .05
,RND*40: GO TO 180
 190 GO TO 50
```

## The ZX81 Listing

```
  10 REM * ZX81 MATCHSTICKS *
  20 LET E=0
  30 LET Z=16+INT (RND*9)
  40 LET H=2+INT (RND*2)
  50 PRINT AT 3,3;"MAXIMUM TO TA
KE IS ";CHR$ (H+156)
  60 IF E>0 THEN PRINT AT 5,4;"Y
OU TOOK ";CHR$ (E+156);TAB 16;"I
 TOOK ";CHR$ (Q+156)
  65 PRINT AT 7,0;
  70 FOR K=1 TO Z
  80 PRINT K;"█ ";
  90 IF RND>.85 THEN PRINT
  95 IF RND>.85 THEN PRINT
 100 NEXT K
 110 PRINT AT 19,0;"HOW MANY WIL
L YOU TAKE?"
 115 INPUT E
 120 IF E>H OR E<1 THEN GOTO 115
 130 CLS
 135 LET Z=Z-E
 140 IF Z=0 THEN PRINT AT 10,10;
"I WIN";END
 150 LET Q=Z-1-INT ((Z-1)/(H+1))
*(H+1)+INT (RND*5)-2
 160 IF Q>Z OR Q<1 OR Q>H THEN G
OTO 150
 170 LET Z=Z-Q
 180 IF Z=0 THEN PRINT AT 10,4;"
I TOOK ";Q;", SO YOU WIN";END
 190 GOTO 50
```

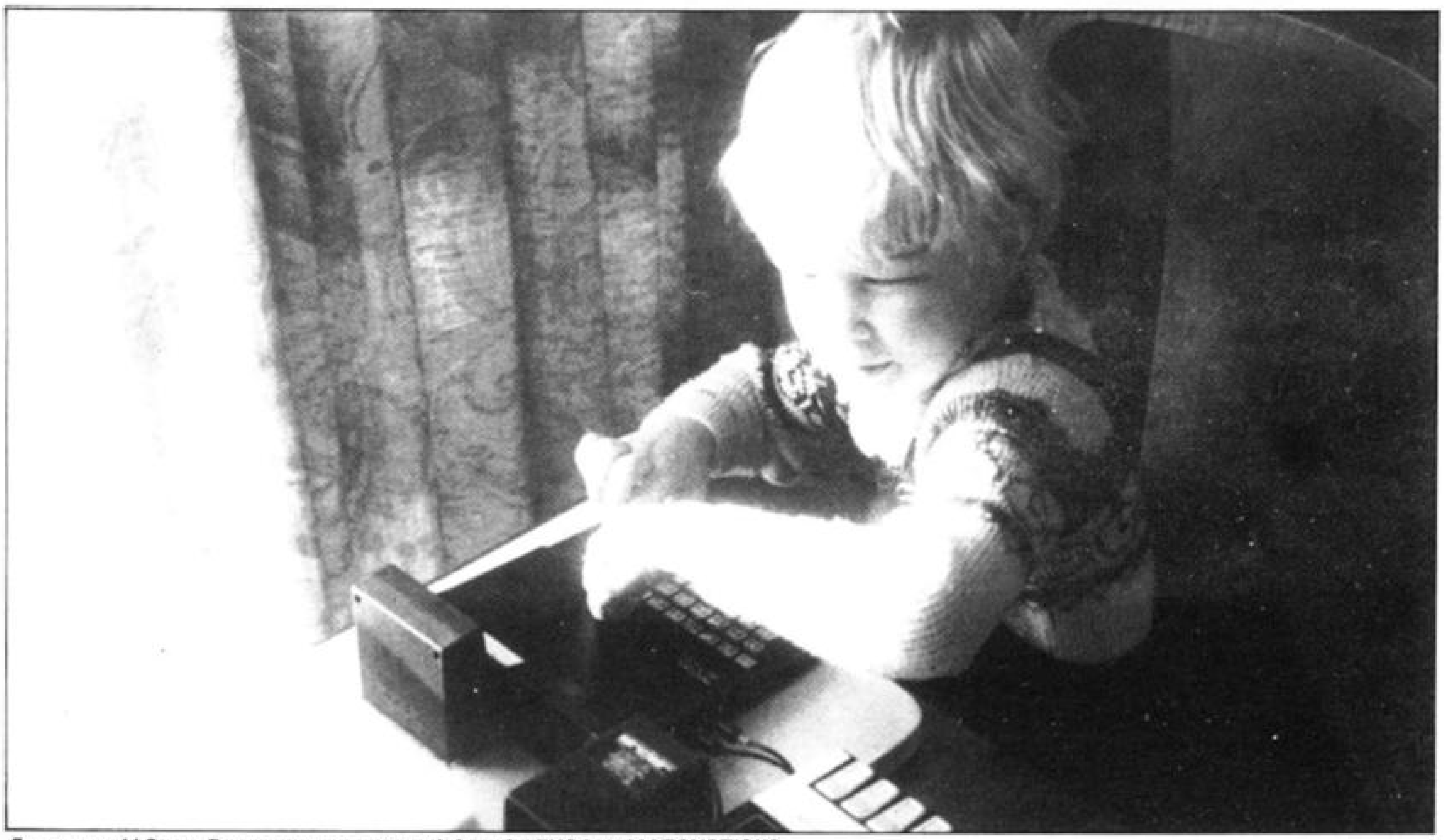
```
    MAXIMUM TO TAKE IS 3

     YOU TOOK 3  I TOOK 3

1█ 2█ 3█ 4█ 5█ 6█
7█ 8█ 9█ 10█
11█ 12█ 13█ 14█
15█

16█

HOW MANY WILL YOU TAKE?
```



*Four-year-old Steve Dommett attempts to defeat the ZX81 at MATCHSTICKS.*

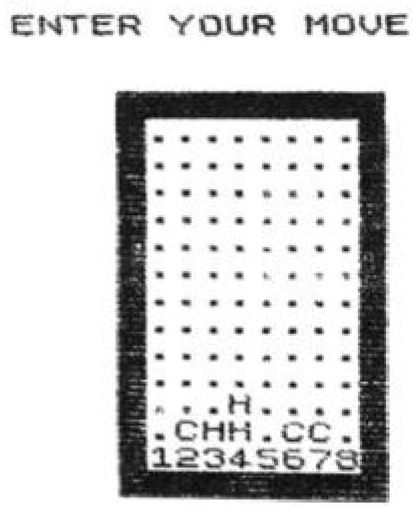
# The ZX81 gets its thinking cap on to challenge a mere human in its own version of 'Four in a Row', or 'Connect Four'.

## Four in a Row

A playing board like that shown in the sample printout is displayed on the screen. You are the letter "H", the computer is the letter "C". You enter the number at the bottom of the row in which you wish to appear, and your piece will be printed there. The program is fairly slow, and has been designed to be run in the FAST mode. If you'd prefer to run it in SLOW, and the board printout (from line 100) looks good in action in SLOW, delete lines 8 and 1006.

There is no mechanism to know when the game is over, nor who has won. You may like to add one once you understand how the program works.

*FOUR IN A ROW display.*




*Program listing.*

```
   1 REM     FOUR IN A ROW
   7 REM
   8 FAST
  10 GOTO 5000
  20 LET E=C
  30 LET E=E+N
  35 IF E<1 THEN RETURN
  40 IF A(E)<>T THEN RETURN
  50 LET K=K+1
  60 GOTO 30
  90 REM ************
 100 PRINT AT 6,10;"██████████"
 110 FOR A=12 TO 1 STEP -1
 120 PRINT AT 19-A,10;"█";
 130 FOR B=2 TO 9
 140 PRINT CHR$ (A(A*10+B));
 150 NEXT B
 160 PRINT "█"
 165 PRINT AT 19-A,10;"█"
 170 NEXT A
 180 PRINT TAB 10;"█12345678█"
 185 PRINT TAB 10;"██████████"
 190 RETURN
 195 REM ************
 200 FOR X=1 TO 4
 210 LET K=0
 220 LET N=X(X)
 230 GOSUB 20
 240 LET N=-N
 250 GOSUB 20
 260 IF K>L THEN LET G=C
 270 IF K>L THEN LET L=K
 280 NEXT X
 290 RETURN
 300 REM ************
1000 GOSUB 100
1005 PRINT AT 3,6;"ENTER YOUR MO
VE"
1006 PAUSE 4E4
1010 LET D=CODE INKEY$-28
1015 IF D=-28 THEN GOTO 1010
1017 PRINT AT 3,6;"
"
1020 IF D<1 OR D>8 OR A(H(D)*10+
D+1)<>CODE "." THEN GOTO 1010
1030 LET Y=CODE "H"
1040 LET C=H(D)*10+D+1
1050 LET H(D)=H(D)+1
1060 LET A(C)=Y
1070 LET L=0
1090 GOSUB 200
1110 REM ************
2000 LET T=0
2010 LET T=T+1
2020 LET P=0
2030 LET Y=CODE "C"
2040 LET D=0
2050 LET D=D+1
2060 LET L=0
2070 IF T=2 THEN LET L=1
2080 IF T=2 THEN LET Y=CODE "H"
2090 LET C=H(D)*10+D+1
2095 IF A(C)<>CODE "." THEN GOTO
 2250
2100 GOSUB 100
2110 LET H=G
2120 IF L>2 THEN GOTO 3010
2130 IF L<2 THEN GOTO 2250
2140 LET M=L
2150 LET L=0
2160 LET Y=CODE "C"
2170 LET C=C+10
2175 IF A(C)<>CODE "." THEN GOTO
 2200
2180 GOSUB 200
2190 IF L>2 THEN GOTO 2250
2200 IF M<P THEN GOTO 2250
2210 IF T=2 AND M<2 THEN GOTO 22
50
2220 LET P=M
2230 LET J=H
2250 IF D<>8 THEN GOTO 2050
2255 IF T<>2 THEN GOTO 2010
2260 IF P<2 THEN GOTO 2500
2270 LET C=J
2280 GOTO 3000
2500 LET D=INT (RND*8)+1
2510 LET C=H(D)*10+D+1
3000 LET D=C-10*INT (C/10)-1
3010 LET A(C)=CODE "C"
3020 LET H(D)=H(D)+1
3030 GOSUB 200
3040 LET Y=CODE "C"
3050 LET L=0
3060 GOSUB 100
3070 IF L>2 THEN GOTO 5000
3080 GOTO 1005
3090 REM ************
5000 DIM A(150)
5010 DIM H(8)
5020 FOR H=1 TO 6
5030 LET H(H)=1
5040 NEXT H
5050 FOR A=1 TO 12
5060 FOR B=2 TO 9
5070 LET A(A*10+B)=CODE "."
5080 NEXT B
5090 NEXT A
5100 DIM X(4)
5110 LET X(1)=1
5120 LET X(2)=9
5130 LET X(3)=10
5140 LET X(4)=11
5150 LET G=0
5160 LET T=0
5200 GOTO 1000
5210 REM ************
```

# Subscriptions

Make sure you get every issue of *ZX Computing* – Now bi-monthly!




Just £11.50 will ensure the next six issues will be lovingly wrapped and posted to you. Just fill in the form below, cut it out and send it with your cheque or postal order (made payable to ASP Ltd) to:-

ZX Computing Subscriptions
513 London Road,
Thornton Heath,
Surrey CR4 6AR

Alternatively you can pay by Access or Barclaycard in which case simply fill in your card number, sign the form and send it off. Do NOT send your card!

Make the most of your ZX computer with *ZX Computing* – Now bi-monthly!

Subscription Order Form

Cut out and SEND TO:

ZX COMPUTING Subscriptions
513, London Road,
Thornton Heath,
Surrey, England.

Please commence my subscription to ZX Computing with the very next issue.

SUBSCRIPTION RATES

(tick – as appropriate)

£11.50 for six issues UK
£13 for six issues overseas surface mail.
£23.80 for six issues overseas airmail.

*I am enclosing my (delete as necessary) Cheque/Postal Order/International Money Order for £..........*
*(made payable to A.S.P. Ltd)*
*OR*
*Debit my Access/Barclaycard**
*(*delete as necessary)*

Please use BLOCK CAPITALS and include post codes.

Name(Mr/Mrs/Miss) ..................................
delete accordingly

Address ..................................

..................................

..................................

Signature ..................................

Date ..................................

# Takin' care of business

**THE FAST ONE, produced by Campbell Systems, is a generalised business filing and reporting system. Ian Logan, author of several outstanding ZX81 books, takes a look at The Fast One — and likes what he finds.**

This program is supplied on a cassette tape accompanied by a detailed ten page instruction leaflet. Initially the user will be overwhelmed as this program is daunting in the extreme. However John Campbell very skillfully introduces his program to the user by supplying on the cassette both the 'master' program and a fine demonstration program.

THE FAST ONE (TFO) is a general filing and reporting system which means that the 'master' program holds an empty file that the user fills with his own specific-task data, whether that be for example — names and addresses of clients and their particulars — or the past success of one's sproting idols, or even — recipes. Initially, therefore, the 'master' program will not unnaturally do next to nothing but the 'demonstration' program will perform impressively from the start.

The essential parts to TFO involve:

i. Filing your data as a series of records.
ii. Formatting your report(s).
iii. Selecting which records are to be reported.
iv. Making the actual report on the TV screen and printer if required.

In the demonstration program the first two steps had already been done in so far that 11 records had been entered and three types of reports formatted. It is then left to the user to select which records are to be reported. For example, it is possible to select from the demonstration program's file of 11 staff records only those records for the members of the staff that *are* over 40 years of age, *do* earn over £6000 and *do not* work in ADMIN. The result of the search can then be displayed, and printed, in file-order, or in two different alphabetically-sorted displays.

The strong point of this program is its immense versatility. The program is predominantly menu-driven and the number of different menus is in itself amazing. The main menu allows the user to add, update or delete records each containing up to 36 fields (items), each of which may have up to 32 characters. However before an actual record can be entered the 'items' that it is to hold have to be defined, ie. NAME to be the name of a member of staff, SALARY to be the salary, etc. Once the records have been entered they will be inaccessible until a report has been defined. The technique for doing this is difficult to explain simply but the user has to describe the format of the whole screen including titles, spaces and the size and position of the data items. The resulting formatting instructions do however, once obtained, look very simple. Included in the formatting procedure is the requirement for the user to specify how the records are to be sequenced. This sequencing can be chronological (file order) or sorted on *any* item, ie. in age order, house number, alphabetic order of colour, etc. Only when all these stages have been passed will the user be able to reproduce the records from his file and if he should wish to make a 'selection' of the kind mentioned earlier.

This program is fantastic in its elegance, it's sheer speed and ease of use. It is a pleasure to use seriously, as well as being an object of study. Indeed the features included in this program embody many aspects of modern file handling and the program is therefore of great educational value.

Needless to say TFO is mainly in machine code and occupies about 5K of RAM when the file is empty. The file is managed dynamically and therfore only the file and the master program are ever saved on the tape. The sheer speed with which records can be manipulated is incredible and this single fact is shown in no better way than to respond 'B' to the main menu when the computer tallies the 'spare bytes' and goes from 00000 to 11700, clocking the bytes one by one — in 2 seconds.

This program is most strongly recommended and I find it the most interesting program that I have yet seen for the ZX81. Campbell Systems are at 15 Rous Road, Buckhurst Hill, Essex. 1G9 6BL.

```
          SORTED BY NAME

                 AGE  SALARY

ATKINS P         29   6500
ADMIN

CARTER H         27   6605
SALES

FARNHAM F.W      45   12000
ADMIN

GALLAGHER        34   7500
SALES

MACKAY A.H.      42   7860
PR

--------DISPLAY/PRINT----------
NLNEXT TTOP +NRECS PPRINT MMENU

          SELECTED=00011
```

```
        UNSORTED TABULAR

NAME                SALARY  AGE

ATKINS P            6500    29
ROBERTS B           7600    46
PICKERING J         5600    26
GALLAGHER           7500    34
TUCKER L            4900    23
FARNHAM F.W         12000   45
PETERS A            7850    55
MARSH J.T           5500    31
MACKAY A.H.         7860    42
UPTON J.J           11000   45
CARTER H            6605    27

--------DISPLAY/PRINT----------
NLNEXT TTOP +NRECS PPRINT MMENU

               SELECTED=00011
```

```
        DOSSIER BY NAME

NAME       GALLAGHER

DEPT       SALES

AGE        34

SALARY     7500

--------DISPLAY/PRINT----------
NLNEXT TTOP +NRECS PPRINT MMENU

          SELECTED=00011
```

```
TOTAL SALARY

00000083115.00

AVERAGE=7555.9091

--------TOTAL/AVERAGE----------
NLMENU

               SELECTED=00011
```

# Keeping tabs on your cheques

The program leads you through entering information regarding your account, gives you the option of altering a particular entry if needed, and (at the end of the run) prints out information on all the cheques written (who it was made out to, and why, and the amount). The program then prints out the final balance and, if necessary, gives you the good news that you are overdrawn.

## This 16K program should help make sure you do not become overdrawn — and if you do, it will tell you.

```
10 REM CHEQUEBOOK BALANCER
20 REM (C) HARTNELL 1982
100 SCROLL
110 PRINT "ENTER THE LAST BALANCE KNOWN"
120 INPUT BAL
125 SCROLL
126 PRINT "STARTING BALANCE $";BAL
127 SCROLL
130 SCROLL
140 PRINT "ENTER, PRESSING RETURN AFTER"
150 SCROLL
160 PRINT "EACH ONE, DEPOSITS MADE SINCE"
165 SCROLL
170 SCROLL
175 PRINT "DEPOSIT","BALANCE"
180 INPUT DEP
190 IF DEP=0 THEN GOTO 240
200 LET BAL=BAL+DEP
210 SCROLL
220 PRINT DEP,BAL
230 GOTO 180
240 SCROLL
250 PRINT "THE BALANCE BEFORE LATEST"
255 SCROLL
260 PRINT "CHEQUES WRITTEN WAS $";BAL
270 SCROLL
280 SCROLL
290 SCROLL
300 PRINT "HOW MANY CHEQUES HAVE YOU"
310 SCROLL
320 PRINT "WRITTEN SINCE THEN?"
330 INPUT NUM
340 DIM A$(NUM,22)
350 DIM B(NUM)
360 FOR G=1 TO NUM
365 SCROLL
370 PRINT "ENTER NAME MADE OUT TO"
380 INPUT N$
390 SCROLL
400 PRINT "ENTER REASON FOR CHEQUE"
410 INPUT M$
420 LET A$(G)=N$+" - "+M$
440 SCROLL
450 PRINT "AND HOW MUCH WAS CHEQUE FOR?"
460 INPUT Q
470 LET B(G)=Q
480 SCROLL
490 PRINT A$(G);" - $";B(G)
495 SCROLL
500 PRINT "IF THIS IS CORRECT, PRESS RET."
510 SCROLL
520 PRINT "IF IT IS NOT, ENTER ""E"""
525 INPUT U$
530 IF U$="E" THEN GOTO 365
540 LET BAL=BAL-Q
545 SCROLL
550 NEXT G
560 SCROLL
570 SCROLL
580 PRINT "THIS IS A RECORD OF YOUR"
590 SCROLL
600 PRINT "CHEQUES TO DATE:"
610 SCROLL
620 SCROLL
630 FOR G=1 TO NUM
640 SCROLL
650 PRINT A$(G);" - $";B(G)
660 NEXT G
670 SCROLL
675 SCROLL
680 PRINT "FINAL BALANCE IS ";BAL
690 SCROLL
700 SCROLL
800 PRINT "ENTER ""R"" TO RUN THE PROGRAM"
810 SCROLL
820 PRINT TAB 8;"FROM SCRATCH"
830 SCROLL
840 PRINT "OR ""B"" TO RUN FROM"
850 SCROLL
860 PRINT TAB 4;"CURRENT BALANCE"
870 SCROLL
880 PRINT "OR ""P"" FOR A DETAILED"
890 SCROLL
900 PRINT TAB 4;"PRINT-OUT OF CHEQUES"
910 SCROLL
920 PRINT TAB 12;"WRITTEN"
930 SCROLL
940 PRINT "OR ""E"" TO END"
950 INPUT U$
960 IF U$="R" THEN RUN
970 IF U$="B" THEN GOTO 125
980 IF U$="P" THEN GOTO 1020
990 IF U$="E" THEN STOP
1000 GOTO 950
1010 REM **PRINT OUT**
1020 FOR G=1 TO NUM
1030 LPRINT A$(G),B(G)
1040 NEXT G
1050 GOTO 690
```

```
ENTER THE LAST BALANCE KNOWN
STARTING BALANCE $1879

ENTER, PRESSING RETURN AFTER
EACH ONE, DEPOSITS MADE SINCE
  DEPOSIT            BALANCE
  230                2109
```

```
35              2144
234             2378
189             2567
42              2609
75.75           2634.75

THIS IS A RECORD OF YOUR
CHECKS TO DATE:

JONES - PLUMBING          - $43.5
SMITH - WINDOWS           - $56.75
TIMBOB - GAMES            - $32
```

```
FINAL BALANCE IS 2552.5

ENTER "R" TO RUN THE PROGRAM
        FROM SCRATCH
OR "B" TO RUN FROM
    CURRENT BALANCE
OR "P" FOR A DETAILED
      PRINT-OUT OF CHECKS

              WRITTEN
OR "E" TO END
```

## VAT Calculation

This is a 1K program which will work out the price-plus-VAT and price-less-VAT of any amount entered in pounds (within the limits of the numbers the ZX81 can handle of course). The rate of VAT is fixed at 15% by the formulae in lines 60 and 90 and these must be changed for any other rate of VAT. The figures displayed are rounded off to the nearest penny and justified to two places of decimal with zeros added as are required by the subroutine starting at line 150.

The program may be of use to shoppers who wish to display VAT inclusive and VAT exclusive. You should know whether the original amount you enter includes or excludes VAT, and choose the figure you require off the screen — both figures are displayed to prevent the user having to specify which answer is wanted. This was found to be a far better method. When the ZX81 asks for an input, enter the figure in pounds, even if it is less than £1.00.

```
VATCALCULATOR, VERSION 1, 1K


  10 PRINT "ENTER AMOUNT (IN POU
NDS) £"
  20 INPUT B
  30 CLS
  40 LET A=B
  50 PRINT ,,"AMOUNT=";
  60 GOSUB 160
  70 PRINT ,,"AMOUNT+VAT";
  80 LET A=B*15/100+B
  90 GOSUB 160
 100 PRINT "AMOUNT-VAT=";
 110 LET A=B*20/23
 120 GOSUB 160
 130 PRINT
 140 PRINT
 150 GOTO 10
 160 LET A$=STR$ (INT (A*100+0.5
)/100)
 170 IF A$(1)="." THEN LET A$="0
"+A$
 180 LET C=LEN A$-LEN STR$ INT V
AL A$
 190 LET A$=A$+(".00"(C+1 TO ))
 200 PRINT "£";A$
 210 RETURN
```

This next program is a version that allows the user to specify the VAT rate at the outset and to change it during the running of the program if desired, and the program also has a scrolling display. When running the program, you will get a prompt to enter the VAT rate.

If it is 15%, enter 15 (NEWLINE). Do not enter the word PERCENT or a 0/0 symbol as this will cause an error. The computer will then tell you how to change the VAT rate if you wish to change the rate. To do this, you enter the letter Z followed by NEWLINE. You may expect this to cause an error since the computer is expecting a numeric INPUT (line 110). However, Z has previously been defined as a variable (line 10).

and B will take the value of Z, in this case a very small number, IE-9 which would never be entered in the normal useage of this program. So if you enter either IE-9 or Z the program recognises this as a signal that you want to change the VAT rate and takes appropriate action. The thing to note about entering the amout to be evaluated is that you should enter the amount in pounds, without the £ symbol, eg. twenty pounds would be entered as 20, twenty pounds and twenty three pence would be 20.23.

It does not matter in the least if you enter any trailing spaces, eg. 20.00 since this will not affect the arithmetic, and all numbers printed are justified to two places of decimal by the subroutine starting at line 270. This program requires 2K to run. The load and go routine at line 9900 is very useful, because the program runs automatically when loaded from tape and is in the mood of "handy utility" that the program was intended.

```
VATCALCULATOR, VERSION 2,>1K


  10 LET Z=1E-9
  20 SCROLL
  30 PRINT AT 21,0;"ENTER VAT RA
TE (PER CENT)"
  40 INPUT V
  45 SCROLL
  50 SCROLL
  55 PRINT [illegible]
[illegible]
  57 SCROLL
  58 SCROLL
  60 PRINT "VAT RATE=";V;" PER C
ENT"
  70 SCROLL
  80 PRINT "ENTER Z TO CHANGE VA
T RATE"
  90 SCROLL
  95 SCROLL
 100 PRINT "AMOUNT=";
 110 INPUT B
 120 IF B=Z THEN GOTO 30
 130 LET A=B
 140 GOSUB 270
 150 SCROLL
 160 SCROLL
 170 PRINT "AMOUNT+VAT=";
 180 LET A=B*V/100+B
 190 GOSUB 270
 200 SCROLL
 210 PRINT "AMOUNT-VAT=";
 220 LET A=(B*100)/(100+V)
 230 GOSUB 270
 240 SCROLL
 250 GOTO 50
 270 LET A$=STR$ (INT (A*100+.5)
/100)
 280 IF A$(1)="." THEN LET A$="0
"+A$
 290 LET C=LEN A$-LEN STR$ INT V
AL A$
 300 LET A$=A$+(".00"(C+1 TO ))
 310 PRINT "£";A$
 320 RETURN
9900 SAVE "VAT CALCULATOR"
9910 RUN
```

# ZX81 Word Processor

## This word processor program will make text neat and tidy before you print it—and gives you the chance to correct mistakes, using a free-moving cursor.

The sample runs show how the program operates. You enter your text (up to 17 lines deep) as a single string, X$. When you have the text in, you press NEWLINE, and the computer will shuffle the words to ensure that none of them are split at the end of a line.

A menu appears with three options: 1 — correct the text; 2 — LPRINT the text; and 3 — to start again. If you decide to correct the text, it will reappear on the screen, with the words "ENTER 1 TO RETURN TO MENU" above it. You use the 5, 6, 7 and 8 keys to move your cursor in the direction indicated by the arrows on those keys, and the cursor moves along the line of text, inverting the letter it is passing over. Once you find a letter which is wrong, such as the "E" in the word WERKS in the sample run, you press "A" and the words ENTER LETTER TO BE SUBSTITUTED appear at the bottom of the screen. You enter your letter, and press NEWLINE, and the inverse incorrect letter will be altered to the letter you've chosen. Pressing "1" at any time will return you from the 'correction phase' to the original menu, and from this menu you can choose "2" to LPRINT the text.

If you want the text printed, the computer searches through the whole of the string, turning any inverse letters back to their non-inverse equivalents. After LPRINTing, you are shown a further menu, which allows you to run the whole program again from scratch, or to terminate the run. Although this program allows you to correct wrongly spelled words, there is no provision to insert text. You may well wish to add this option.

***Segments of a sample sum.***

```
ENTER TEXT
THIS IS A TRIAL RUN TO ILLUSTRAT
E THE TEXT PROGRAM IN ACTION, TO
 SHOW HOW IT WERKS AND TO DEMONS
TRATE HOW IT CAN MAKE TEXT LOOK
NEAT BEFORE PRINTING


THIS IS A TRIAL RUN TO
ILLUSTRATE THE TEXT PROGRAM IN
ACTION, TO SHOW HOW IT WERKS AND
TO DEMONSTRATE HOW IT CAN MAKE
TEXT LOOK NEAT BEFORE PRINTING

ENTER 1 TO CORRECT TEXT,
2 TO LPRINT, 3 TO START AGAIN


ENTER 1 TO RETURN TO MENU
89 E
THIS IS A TRIAL RUN TO
ILLUSTRATE THE TEXT PROGRAM IN
ACTION, TO SHOW HOW IT WERKS AND
TO DEMONSTRATE HOW IT CAN MAKE
TEXT LOOK NEAT BEFORE PRINTING


ENTER LETTER TO BE SUBSTITUTED


ENTER 1 TO RETURN TO MENU
89 E
THIS IS A TRIAL RUN TO
ILLUSTRATE THE TEXT PROGRAM IN
ACTION, TO SHOW HOW IT WORKS AND
TO DEMONSTRATE HOW IT CAN MAKE
TEXT LOOK NEAT BEFORE PRINTING

ENTER 1 TO CORRECT TEXT,
2 TO LPRINT, 3 TO START AGAIN
```

```
  10 REM WORD PROCESSOR
  15 REM N GOODWIN, T HARTNELL
  20 PRINT "ENTER TEXT"
  30 INPUT X$
  32 LET X$=X$+"
"
  35 CLS
  40 GOSUB 1000
  50 PRINT X$
  60 PRINT
  70 PRINT "ENTER 1 TO CORRECT T
EXT.          2 TO LPRINT, 3 TO ST
ART AGAIN"
  80 IF INKEY$="" THEN GOTO 80
 100 IF INKEY$="3" THEN RUN
 110 IF INKEY$="2" THEN GOTO 400
0
 120 IF INKEY$="1" THEN GOTO 200
0
 130 GOTO 80
1000 REM STOPS WORD SPLITTING
1010 LET N=1
1020 GOSUB 1180
1030 LET N=N+33
1040 IF N>=LEN X$ THEN RETURN
1045 REM SINGLE SPACE IN
          NEXT LINE
1050 IF X$(N)=" " THEN GOTO 1160
1060 GOSUB 1180
1065 REM SINGLE SPACE IN
          NEXT LINE
1070 IF X$(N)=" " THEN GOTO 1030
1080 LET J=0
1090 GOSUB 1180
1100 LET J=J+1
1105 REM SINGLE SPACE IN
          NEXT LINE
1110 IF X$(N)>" " THEN GOTO 1090
1120 FOR N=N TO N+J-1
1125 REM SINGLE SPACE IN
          NEXT LINE
1130 LET X$=X$(1 TO N)+" "+X$(N+
1 TO )
1140 NEXT N
```

```
1150 GOTO 1030
1160 LET X$=X$(1 TO N-1)+X$(N+1
TO )
1170 GOTO 1020
1180 LET N=N-1
1190 RETURN
2000 REM **CORRECTION**
2010 CLS
2020 PRINT "ENTER 1 TO RETURN TO
 MENU"
2030 LET A=1
2035 PRINT AT 2,0;X$
2040 IF INKEY$="" THEN GOTO 2040
2050 IF INKEY$="8" AND A<LEN X$
THEN LET A=A+1
2055 IF INKEY$="6" AND A<LEN X$+
32 THEN LET A=A+32
2060 IF INKEY$="5" AND A>1 THEN
LET A=A-1
2065 IF INKEY$="7" AND A>32 THEN
 LET A=A-32
2070 IF INKEY$="1" THEN GOTO 70
2075 IF INKEY$="A" THEN GOTO 300
0
2076 PRINT AT 1,0;A;" ";X$(A);"
 "
2080 IF CODE X$(A)<128 THEN LET
X$=X$( TO A-1)+CHR$ (CODE X$(A)+
128)+X$(A+1 TO )
2085 IF A=1 THEN GOTO 2035
2090 IF A>1 AND CODE X$(A-1)>127
 THEN LET X$(A-1)=CHR$ (CODE X$(
A-1)-128)
2092 IF A<32 THEN GOTO 2100
2095 IF CODE X$(A-32)>127 THEN L
ET X$(A-32)=CHR$ (CODE X$(A-32)-
128)
2100 IF A<LEN X$-1 AND CODE X$(A
+1)>127 THEN LET X$(A+1)=CHR$ (C
ODE X$(A+1)-128)
2102 IF A<LEN X$-32 THEN GOTO 20
35
2105 IF CODE X$(A+32)>127 THEN L
ET X$(A+32)=CHR$ (CODE X$(A+32)-
128)
2110 GOTO 2035
3000 REM INSERT CORRECTION
3005 PRINT AT 19,0;"ENTER LETTER
 TO BE SUBSTITUTED"
3010 INPUT H$
3020 LET X$(A)=H$
3025 PRINT AT 19,0;"
 "
3030 GOTO 2035
4000 REM REMOVE INVERSE, LPRINT
4010 FOR G=1 TO LEN X$
4020 IF CODE X$(G)>127 THEN LET
X$(G)=CHR$ (CODE X$(G)-128)
4030 NEXT G
4040 LPRINT X$
4050 CLS
4060 PRINT "ENTER 1 TO RUN AGAIN
"
4070 PRINT TAB 5;"2 TO END"
4080 IF INKEY$="" THEN GOTO 4080
4090 IF INKEY$="1" THEN RUN
```

# Tote that barge, read that data

**Chalfont St Giles in Buckinghamshire is the home of Martin Frobisher, who feeling lost on his ZX81 without the use of READ and DATA, decided to create a routine for it.**

Martin writes to *ZX Computing:* "Having felt lost on the ZX81 without READ/DATA statements, I set about writing a subroutine to solve this problem. When using this subroutine, it is necessary to use GOSUB 9950 in place of READ A. The subroutine is written for numbers only, but can be converted easily to deal with string information. The string which holds the data is placed at the beginning of the program, and the subroutine at the end. Note that there must be a comma at the very end of the 'DATA' within the quote marks of the string in line 10."

The READ/DATA routine is given in program one, along with a simple demonstration program.



***Program One:***

```
   5 REM READ/DATA ROUTINE
   6 REM BY MARTIN FROBISHER
  10 LET A$="42,75,6,333,29,"
  20 LET X=1
  30 LET Y=X
  40 DIM F(5)
  50 FOR G=1 TO 5
  60 GOSUB 9950
  70 LET F(G)=A
  80 NEXT G
  90 FOR G=1 TO 5
 100 PRINT G;" ";F(G)
 110 NEXT G
 120 STOP
9950 LET Y=Y+1
9960 IF A$(Y)<>"," THEN GOTO 995
0
9970 LET A=VAL A$(X TO Y-1)
9980 LET X=Y+1
9990 RETURN
```

```
1 42
2 75
3 6
4 333
5 29
```

The variables used are as follows;

| | | |
|---|---|---|
| MENU | V$ — TITLE (Inverse video subroutine.)<br>K$ — 'Do you want to continue?'<br>G — Number of gold blocks (0) | |
| CHOICE | B$ — AHEAD  D — DOOR<br>L$ — LEFT  T — TUNNEL<br>R$ — RIGHT  C — CAVE | (Choose with A$.) |
| DOOR | Q — Gold blocks<br>C$ — YES/NO — 'Would you like to cross the lake?'<br>K — 'Random' chance of survival. | |
| TUNNEL | * | |
| CAVE | Q — Gold blocks<br>C$ — YES/NO — 'Would you like to investigate the noises?' | |
| INVERSE VIDEO SUBROUTINE | V$ — ZX80 only | |

When the computer prints '> >' this means 'Do you want to continue?'
If 'YES' then simply press NEWLINE
If 'NO' then type SPACE then NEWLINE

```
   1 REM DRAGONS GOLD
   2 REM BY D C OWEN 1982
   3 RAND
   5 LET G=0
  50 SCROLL
  51 SCROLL
  52 SCROLL
  53 SCROLL
  55 PRINT TAB 8;"DRAGON/S GOLD"
  56 SCROLL
  57 SCROLL
  58 SCROLL
  60 PRINT "YOU HAVE:-"
  65 SCROLL
  70 PRINT G;" BLOCKS OF GOLD"
  80 SCROLL
  90 SCROLL
 100 PRINT "AHEAD OF YOU IS A";
 120 GOSUB 1000
 125 LET B$=A$
 127 SCROLL
 130 PRINT "ON THE LEFT IS A";
 140 GOSUB 1000
 145 SCROLL
 150 LET L$=A$
 155 PRINT "AND ON THE RIGHT IS 
A";
 160 GOSUB 1000
 165 LET R$=A$
 170 SCROLL
 171 SCROLL
 172 PRINT "WHICH WAY DO YOU WAN
T TO GO?"
 173 SCROLL
 174 SCROLL
 175 PRINT "A - AHEAD"
 176 SCROLL
 177 PRINT "L - LEFT"
 178 SCROLL
 179 PRINT "R - RIGHT"
 180 INPUT K$
 185 SCROLL
 186 SCROLL
 190 IF K$="A" AND B$="D" THEN G
OSUB 2000
 200 IF K$="R" AND R$="D" THEN G
OSUB 2000
 210 IF K$="A" AND B$="T" THEN G
OSUB 3000
 220 IF K$="L" AND L$="T" THEN G
OSUB 3000
 230 IF K$="R" AND R$="T" THEN G
OSUB 3000
 240 IF K$="A" AND B$="C" THEN G
OSUB 4000
 250 IF K$="L" AND L$="C" THEN G
OSUB 4000
 260 IF K$="R" AND R$="C" THEN G
OSUB 4000
 270 IF NOT (K$="L" OR K$="R" OR
 K$="A") THEN GOTO 170
 280 GOTO 50
 990 REM *************
1000 GOTO 1000+INT (RND*3+1)*100
1100 PRINT " DOOR"
1110 LET A$="D"
1120 RETURN
1200 PRINT " TUNNEL"
1210 LET A$="T"
1220 RETURN
1300 PRINT " CAVE"
1310 LET A$="C"
1320 RETURN
1999 REM ************
2000 REM ** DOOR **
2010 GOTO 2000+INT (RND*4+1)*100
2100 PRINT "IT IS LOCKED. MOVE O
N"
2110 RETURN
2200 LET Q=INT (RND*9+1)*100
2210 PRINT "IT WILL OPEN. THERE"
2215 SCROLL
2220 PRINT "ARE ";Q;" GOLD BLOCK
S IN HERE"
2230 LET G=G+Q
2240 RETURN
2300 PRINT "THERE IS A LAKE HERE
. YOU"
2305 SCROLL
2310 PRINT "CANNOT SEE THE FAR S
IDE."
2315 SCROLL
2320 PRINT "ARE YOU GOING TO TRY
"
2330 SCROLL
2340 PRINT "AND CROSS IT?"
2345 SCROLL
2350 INPUT C$
2360 CLS
2370 IF CODE (C$)<>CODE "Y" THEN
 RETURN
2380 LET K=INT (RND*3)+1
2381 SCROLL
2382 IF K=2 THEN PRINT "YOU HAVE
 ESCAPED WITH"
2383 SCROLL
2384 IF K=2 THEN PRINT "WITH ";G
;" BLOCKS OF GOLD"
2386 IF K<>2 THEN PRINT "UNFORTU
NATELY, YOU HAVE"
2388 SCROLL
2390 IF K<>2 THEN PRINT TAB 10;"
DROWNED..."
2395 STOP
2400 LET K=INT (RND*9+1)*50
2405 SCROLL
2410 PRINT "THIS ROOM CONTAINS A
 DRAGON"
2415 SCROLL
2420 PRINT "IT DEMANDS ";K;" GOL
D BLOCKS"
2425 SCROLL
2430 PRINT "OR IT WILL EAT YOU"
2440 FOR J=1 TO 20
2450 SCROLL
2460 PRINT TAB J;"STAND BY"
2470 NEXT J
2475 SCROLL
2480 IF G>K-1 THEN PRINT "YOU HA
VE ENOUGH"
2485 SCROLL
2490 IF G<K THEN PRINT "...BUT Y
OU HAVEN/T GOT"
2495 SCROLL
2500 IF G<K THEN PRINT "ENOUGH..
.SO BYE BYE";END
2510 LET G=G-K
2520 RETURN
2999 REM **********
3000 REM ** TUNNEL **
3010 IF RND>0.85 THEN RETURN
3015 SCROLL
3020 PRINT "YOU HAVE ESCAPED"
3025 SCROLL
3030 PRINT "WITH ";G;" GOLD BLOC
```

```
KS"
3040 STOP
3999 REM ************
4000 REM  ** CAVE **
4005 SCROLL
4010 GOTO 4000+INT (RND*3+1)*100
4100 PRINT "THE CAVE IS EMPTY."
4105 SCROLL
4110 PRINT TAB 8;"MOVE ON"
4120 RETURN
4200 LET Q=INT (RND*10+1)*100
4210 PRINT "THERE ARE ";Q;" GOLD
 BLOCKS"
4215 SCROLL
4220 PRINT "HERE TO ADD TO YOUR
STORE"
4230 LET G=G+Q
4240 RETURN
4300 IF RND>0.9 THEN GOTO 4400
4301 FOR H=1 TO 24
4305 SCROLL
4307 NEXT H
4310 PRINT "OH NO"
4320 FOR J=1 TO 15
4330 PRINT TAB 2*J;"■■■"
4340 NEXT J
4345 SCROLL
4350 PRINT "IT IS A MINESHAFT...
"
4355 SCROLL
4360 PRINT "YOU ARE DEAD"
4370 STOP
4400 SCROLL
4405 PRINT "THERE ARE NOISES AHE
AD"
4407 SCROLL
4410 PRINT "DO YOU WANT TO INVES
IGATE?"
4420 INPUT K$
4430 IF CODE K$<>CODE "Y" THEN R
ETURN
4440 GOTO 4000
```

## Fastermind

This version of Mastermind (a trade name owned by Invicta), uses the letters A to F. The first listing is for the ZX80, and the second for the ZX81.

```
100 DIM A(4)
110 DIM B(4)
120 PRINT "FASTERMIND A B C D E F"
130 FOR I=1 TO 4
140 LET B(I)=RND(6)
150 NEXT I
160 LET L=0
170 LET L=L+1
180 PRINT
185 PRINT L,
190 LET K=0
195 LET J=0
200 INPUT A$
210 FOR I=1 TO 4
220 LET A(I)=CODE(A$)-37
225 IF A(I)>6 THEN GOTO 180
230 PRINT CHR$(A(I)+165);" ";
235 LET B(I)=ABS(B(I))
240 IF NOT A(I)=B(I) THEN GOTO 280
250 LET K=K+1
260 LET A(I)=0
270 LET B(I)=-B(I)
280 LET A$=TL$(A$)
290 NEXT I
300 FOR H=1 TO 4
310 FOR I=1 TO 4
320 IF NOT A(H)=B(I) THEN GOTO 360
330 LET J=J+1
340 LET B(I)=-B(I)
350 GOTO 370
360 NEXT I
370 NEXT H
400 PRINT ,"";"";
410 IF K=0 THEN GOTO 450
420 FOR I=1 TO K
430 PRINT "* ";
440 NEXT I
450 IF J=0 THEN GOTO 490
460 FOR I=1 TO J
470 PRINT "+ ";
480 NEXT I
490 IF K<4 THEN GOTO 170
READY
```

As you'll see when you run the program, a correct letter in the wrong position will give a "+", while a correct letter in the correct place gives a "*". You are, of course, trying to get four *'s in as short a number of guesses as possible. Note that letters may be repeated within the code. Invalid guesses are rejected. Here's the ZX81 version:

```
 80 REM FASTERMIND
 90 REM BY D C OWEN
100 DIM A(4)
110 DIM B(4)
130 FOR Z=1 TO 4
140 LET B(Z)=INT (RND*6)+1
150 NEXT Z
160 LET L=0
165 SCROLL
168 PRINT "FASTERMIND A B C D E
F"
170 LET L=L+1
180 SCROLL
182 SCROLL
185 PRINT "ENTER GUESS NUMBER ";
L
190 LET K=PI-PI
195 LET J=K
200 INPUT A$
205 SCROLL
210 FOR Z=1 TO 4
220 LET A(Z)=CODE (A$)-37
225 IF A(Z)>6 THEN GOTO 180
230 PRINT CHR$ (A(Z)+165);" ";
235 LET B(Z)=ABS B(Z)
240 IF A(Z)<>B(Z) THEN GOTO 280
250 LET K=K+1
260 LET A(Z)=0
270 LET B(Z)=-B(Z)
280 LET A$=A$(2 TO )
290 NEXT Z
300 FOR H=1 TO 4
310 FOR Z=1 TO 4
320 IF A(H)<>B(Z) THEN GOTO 360
330 LET J=J+1
340 LET B(Z)=-B(Z)
350 GOTO 370
360 NEXT Z
370 NEXT H
400 PRINT TAB 10;"SCORED ";
410 IF K=0 THEN GOTO 450
420 FOR Z=1 TO K
430 PRINT "* ";
440 NEXT Z
450 IF J=0 THEN GOTO 490
460 FOR Z=1 TO J
470 PRINT "+ ";
480 NEXT Z
490 IF K<4 THEN GOTO 170
500 PRINT "YOU DID IT"
```

```
FASTERMIND A B C D E F

ENTER GUESS NUMBER 1
A B C D     SCORED + +

ENTER GUESS NUMBER 2
E F C D     SCORED * + +

ENTER GUESS NUMBER 3
C B C C     SCORED * +

ENTER GUESS NUMBER 4
A B B E     SCORED *

ENTER GUESS NUMBER 5
```

**MELBOURNE HOUSE • ZX81 • SPECTRUM**

# Why is this man smiling?



You'd be smiling too if you were Dr. Ian Logan.

Dr. Logan is shown receiving the Rosetta Stone Award for his perceptive insights into the way the ZX81 ROM operates. Melbourne House are proud to be the publishers of Dr. Logan's books.

Melbourne House are Sinclair specialists, publishing books that are winners in every category. Whether you are a beginner or an experienced programmer, you'll need Melbourne House books.

If you've got a Spectrum on order (or better yet, if you've already got your Spectrum), then you'll be glad to know that Dr. Logan is working on **Understanding Your Spectrum**, sure to become the definitive book on the Spectrum.

So if you want to smile like Dr. Logan, you can do one of two things: Order one of our books from your bookseller or fill in the coupon and really understand your Sinclair, or you can write a great book or program and send it to us. Who knows? Your smile may be in this ad. next year!

- O **Machine Language Made Simple For Your Sinclair**, £8.95
- O **Understanding Your ZX81 ROM**, by Dr. Ian Logan, £8.95
- O **ZX81 ROM Disassembly Part A**, by Dr. Ian Logan, £7.00
- O **ZX81 ROM Disassembly Part B**, by Dr. Ian Logan and Dr. Frank O'Hara, £8.00
- O **Not only 30 Programs for the ZX81: 1K**, £6.95
- O **The Complete Sinclair ZX81 Basic Course**, £17.50

Orders to **Melbourne House Publishers**, 131 Trafalgar Road, Greenwich London SE10
Correspondence to Glebe Cottage, Station Road, Cheddington, Leighton Buzzard, BEDS LU7 7NA

☐ Please rush me your new Spectrum/ZX81 catalogue. (Please specify.)
☐ Please send me books as marked above. Note: Please add 80p for post, pack & VAT.

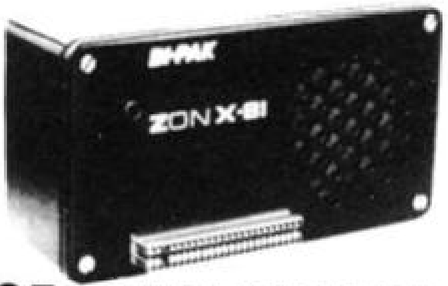
Name ........................................ ZXC

Address ........................................ Postcode ..........

**MELBOURNE HOUSE • ZX81 • SPECTRUM**

# SOUND with ZX-81!

## MAKE AMAZING SOUND EFFECTS WITH YOUR ZX-81




## £25.95 THE ZON X-81

incl p&p & VAT

- The ZON X-81 SOUND UNIT is completely self-contained and especially designed for use with the ZX-81. It just plugs in – no dismantling or soldering
- No power pack, batteries, leads or other extras.
- Manual Volume Control on panel – ample volume from built-in loudspeaker
- Standard ZX-81 – 16K Rampack or printer can be plugged into ZON X-81 Sound Unit without affecting normal ZX-81 operation
- Huge range of possible sounds for games or Music, Helicopters, Sci-Fi, Space Invaders, Explosions, Gun-shots, Drums, Planes, Lasers, Organs, Bells, Tunes, Chords etc. or whatever you devise!
- Uses 3 channel sound chip giving programme control of pitch, volume of tones and noise, all with envelope control
- Easily added to existing games or programmes using a few simple 'BASIC' lines

FULL instructions with many examples of how to obtain effects and the programmes, supplied. Fully Guaranteed

**BI-PAK** Dept SU/P.O. Box 6, 63A, High Street, Ware, Herts.

Access & VISA accepted Ring 0920 3182 for immediate despatch.

Measurement and Control with your

# ZX81 SPECTRUM

and our proven ANALOGUE PORT

**Already many in use in industry, education, hospitals labs., agriculture and the home. Now extremely low cost control operations and even robotics become a reality.**

**The ANALOGUE PORT plugs directly into your ZX81 and offers**

- **8 separate analogue voltage measuring inputs**
- **8 switching outputs for relays, LEDs, sound output**
- **Amplifier for mV sensitivity on one input channel**
- **6 control lines for further expansion**
- **Stackable connector for RAM pack, printer**
- **Self contained, no extra power supply etc. required**

**Thermometers, light sensors, microphones, joysticks . . . in fact anything which produces a varying voltage, resistance or current may be connected directly to your ZX81 via this port. A comprehensive manual is included which assumes no previous expertise.**

only **£29.95** ready built and tested

*** Return of post delivery ***

ZX Frequency Analysis software (Fast Fourier Transform) on cassette and with manual **£15.20**

ZX Experimenter Kit 23 + 23 edge connector, stacking, strip veroboard + suggested circuits **£6.50**

Acorn Atom Eprom programmer, complete System, ready to go including full manual **£55** (28 Days Delivery)

ZX Input – Output Port **£16.50**

**Make cheques payable to:**
**UNIVERSITY COMPUTERS, 5 ST. BARNABAS ROAD CAMBRIDGE. Delivery return of post on all items (except EPROM prog.). Official, trade, bulk orders accepted. Please add £1 p&p to cost of order.**

# Coaxing a Rainbow from your Spectrum

## The Spectrum, as we all know, is a colour computer. But how easy is the colour to use, and how effective is it? Tim Hartnell attempts to answer those questions, and shows you how to program your new ZX Spectrum . . . in colour.




The Spectrum has eight colours (if you count black and white), which are coded from zero to seven. The colours and their numerical codes are:

0 — black
1 — blue
2 — red
3 — magenta (purple)
4 — green
5 — cyan (pale bluey-green)
6 — yellow
7 — white

The lower the number, the darker the colour. On a black and white set the lower numbers are closer to black, the higher numbers to white.

You can colour the PRINT output using the INK statement the background using the PAPER statement, and the border with BORDER. Running program one will show the colours in action. Notice that CLS is used after the PAPER colour is defined (in line 70). This is to ensure that the whole screen area turns that colour. Leave it out and see what happens.

You can use the program colours directly in a program with good effect, as program two — COLOUR CODE — shows. This is a variation of 'Mastermind' but, as you'll see by running it, the program expects you to guess a code of four colours, not four numbers or letters as in most computer versions of the game. Enter and run the game, then return to this article for an explanation of the Spectrum colour and graphics commands which are used in it.

Line 20 (POKE 23609, 100) changes the rate of 'click' when you press a key into a beep, to act as positive feedback when you press a key. I tend to use this all the time, and find it very useful when programming. Line 80 sets the ink and border black (0) and the paper white (7). The routine from lines 100 to 120 print out the six colours (printing a blob of each colour) in a diagonal line, with the numbers next to the colour they refer to. Line 150 waits until any key is pressed before continuing.

The routine from 220 to 300 picks the colours, making sure that all four are different. Line 210, meanwhile, has moved the print position down one (using the apostraphe from the 7 key, accessed with the red shift key), and lines 180 to 200 have printed the six colours

```
430 LET B=B+1: BEEP .2,B*15
440 LET G(Z)=0
450 NEXT Z
460 FOR Z=1 TO 4
470 IF G(Z)=0 THEN GO TO 520
480 FOR J=1 TO 4
490 IF C(Z)<>G(J) THEN GO TO 51
0
500 LET W=W+1: BEEP .2,60-B*15
510 NEXT J
520 NEXT Z
530 FOR T=4 TO 1 STEP -1
540 PRINT INK H(T);"█ ";
550 NEXT T
560 PRINT INK 0;"";B;" BLACK";
570 IF B<>1 THEN PRINT "S";
580 PRINT " AND ";W;" WHITE";
590 IF W<>1 THEN PRINT "S"
600 IF W=1 THEN PRINT
610 IF B=4 THEN PRINT "YOU GOT
IT, IN ";G;" GUESS";
620 IF G>1 AND B=4 THEN PRINT "
ES"
630 IF B<>4 THEN NEXT G
650 PRINT ''"THE CODE WAS ";
660 FOR H=1 TO 100: NEXT H
670 FOR T=4 TO 1 STEP -1
680 FOR H=1 TO 50: NEXT H
690 BEEP .2,T*10: PRINT INK C(T
);"█ ";
700 NEXT T
710 FOR H=1 TO 60: BEEP .01,H:
NEXT H
720 POKE 23692,-1
```

```
10 REM COLOUR CODE
20 POKE 23609,100
30 DIM C(4)
40 DIM G(4)
50 DIM H(4)
60 INK 0: BORDER 0: PAPER 7: C
LS
70 PRINT '''TAB 3;"I AM THINKI
NG OF A 4-COLOUR"
80 PRINT '"CODE. YOU HAVE 10 G
UES TO GUESS"
90 PRINT '"IT. I CHOOSE FROM T
HESE COLOURS"''
100 FOR C=1 TO 6
110 PRINT TAB 4+C; INK 0;C;" >
"; INK C;"█"
120 NEXT C
130 PRINT '" ALL 4 COLOURS ARE
DIFFERENT."
730 PRINT ''"DO YOU WANT ANOTHE
R GAME?"
735 PRINT TAB 8;"ENTER Y OR N"
740 LET A$=INKEY$: IF INKEY$=""
THEN GO TO 740
750 IF CODE A$<>CODE "N" THEN R
UN
760 CLS
770 PRINT '' INK RND*6;TAB RND*
15;"OK, BYE FOR NOW!"
780 POKE 23692,-1
790 FOR H=1 TO 25
800 NEXT H
810 GO TO 770
```

across the top of the screen, together with the numbers which refer to them.

Line 310 starts the loop to give 10 guesses. The second half of line 310 (POKE 23692, −1) ensures that if the screen is ever filled, it will automatically scroll, without requiring a response to the question "scroll?" which you often otherwise get at the bottom of the screen. Along with the key press beep, this automatic scroll POKE is something I use in just about every Spectrum program.

Line 320 asks for the guess to be entered, and once it has (line 330), uses the backspace (CHR$8) 32 times to back over the line requesting the entry of the guess. Line 320 overprints this with blanks. This means that the line ENTER GUESS 2 is erased, but previous guesses (and the colour code at the top) are not, so you can look at previous guesses to help you work out your answer. You enter your guess, by the way, by entering a four-digit number, using the colour code given at the top of the screen. That is, to enter BLUE just press 1.

The routine from lines 350 to 390 strip the number you have entered down to four separate digits, the variables for blacks (B) and whites (W) are set to zero in line 400, and then the guess is compared with the four-digit code the computer has thought of, giving little beeps for 'whites' or 'blacks' as it finds them. If you are right, the program tells you. If you are not, and you have not used up your ten guesses, you are told of the digits of the right colour in the right position 'blacks' as it finds them. If you in the wrong position (whites) and given another guess.

Once you've followed the explanation through (and SAVEd the program if you want to), erase it using NEW and enter our next program (program three) to discover some other applications of Spectrum graphics commands.

You will know that you can use PRINT AT 4, 7; "TEST" to print the word TEST four lines down, and starting seven spaces across. The control character CHR$ 22 behaves like PRINT AT, but with a difference. To get the same result as PRINT AT 4, 7; "TEST" you need to enter PRINT CHR$22 + CHR$ 4 CHR$ 7; "TEST".

However, because the ZX Spectrum allows concatenation (the adding together of strings), you can add all these CHR$'s to equal one string. This can be quite useful, if you wish to specify a particular PRINT AT location several times in a program. Run program three, and you'll see this working.

TAB can be emulated by preceding CHR$ n, where n is the number of spaces (plus one) you wish to start printing on a line, with CHR$ 23. Run program four to see this in action. However, as CHR$ 23 really expects to be followed by two numbers (n and m, which has the same effect as PRINT TAB n + 256*m), you can precede the information within the quote marks with a space, or a dummy letter (X in our example), which will not be printed. Run program four and you'll see that instead of printing XTEST right down the screen, it will simply print TEST.

At the start of this article we discussed the eight colours and looked at how these could be used for the information which is printed (INK), the background (PAPER) or the border (BORDER). The information printed can be modified by the use of two additional commands, BRIGHT and FLASH. Program five shows these in action. Enter and run it, then return to this article for a brief discussion on these two new statements. Although the effect of flashing is impossible to miss, you may need to look a little more closely to see the effect of BRIGHT. Once you have run this program, look at the word BRIGHT, just under NORMAL near the top of the screen. You'll see this is a different shade of green. The white on green (the sixth line down on the screen) shows the effect of BRIGHT more clearly. Compare the 'lightness' of the word BRIGHT here with the word FLASHING just above it. With the non-flashing words printed in green on red (a pretty awful combination), you'll see that the 'bright' word is somewhat easier to read than is the 'normal' one.

Although the numbers zero

```
140 PRINT '''"   PRESS ANY KEY TO BEGIN..."
150 PAUSE 4E4
160 CLS
170 PRINT AT 1,5;
180 FOR C=1 TO 6
190 PRINT INK 0;C;">"; INK C;"█";
200 NEXT C
210 PRINT '
220 LET C(1)=INT (RND*6)+1
230 LET Z=1
240 LET Z=Z+1
250 LET C(Z)=INT (RND*6)+1
260 LET J=0
270 LET J=J+1
280 IF C(J)=C(Z) THEN GO TO 230
290 IF J<Z-1 THEN GO TO 270
300 IF Z<4 THEN GO TO 240
310 FOR G=1 TO 10: POKE 23692,-1
320 PRINT INK 0;"ENTER GUESS NUMBER ";G
330 INPUT A
335 FOR Q=1 TO 32: PRINT CHR$ 8;: NEXT Q
340 PRINT "
"
350 FOR Z=1 TO 4
360 LET G(Z)=A-10*INT (A/10)
370 LET H(Z)=G(Z)
380 LET A=INT (A/10)
390 NEXT Z
400 LET B=0: LET W=0
410 FOR Z=1 TO 4
420 IF C(Z)<>G(Z) THEN GO TO 450
```

```
 I AM THINKING OF A 4-COLOUR
CODE. YOU HAVE 10 GOES TO GUESS
IT. I CHOOSE FROM THESE COLOURS
      1 >>█
       2 >>█
        3 >>█
         4 >>█
          5 >>█
           6 >>█
 ALL 4 COLOURS ARE DIFFERENT.

  PRESS ANY KEY TO BEGIN...
```

```
10 REM PROGRAM ONE
20 REM COLOUR DEMONSTRATION
30 FOR B=0 TO 7
40 FOR P=0 TO 7
50 FOR I=0 TO 7
60 BORDER B
70 PAPER P: CLS
80 INK I
90 PRINT AT 10,10;"BORDER ";B;TAB 10;"PAPER ";P;TAB 10;"INK ";I
100 FOR U=1 TO 60: NEXT U: BEEP .15,P*I+2*B
110 NEXT I
120 NEXT P
130 NEXT B
```

to seven have been explained for INK, PAPER and BORDER, other numbers can be used. Using 8 (as in PAPER 8) means that no matter which is printed at this point, the colour will remain unchanged. This is not particularly useful in ordinary programming, but the number 9 can be quite effective. The '9' means contrast, and ensures that if you are printing on a light background, it will print the words in black, and in white on a dark background, somewhat like the way the colour of an INPUT statement changes depending on the border colour. Program six shows this in action printing randomly-generated letters of the alphabet in random positions on the screen, against a randomly chosen paper colour. Run program six for a while to see this and then return to this article for our next useful graphics command.

The word OVER is very useful, and can produce some very odd effects. You will have noticed an apparently useless line at the end of program six (line 140). Using the edit control, put this line in place of line 110, and change the 32 at the end of line 100 into 300. You'll notice from time to time that letters are printed on top of a letter which had previously been printed in that position. The OVER command means that the new letter does not wipe out the one below it, but simply compliments it from the other to form a new shape. This allows us to build up some characters of our own. Enter and run program seven to create some of your own. It is very hard to predict the effect of 'adding' various letters in this way. For example, a small "o" and a small "w" combine to produce what appears to be a capital "T".

You'll remember we discussed the way CHR$ 22 and CHR$23 could be used to replace PRINT AT and TAB, and the way these can be added together (concatenation) so that the whole command can be held in a single string. The same can be done with other commands. The control characters, and the commands they replace, are: CHR$ 16 – INK; CHR$ 17 – PAPER; CHR$ 18 – FLASH; CHR$ 19 – BRIGHT; CHR$ 20 – INVERSE; CHR$ 21 – OVER. These are followed by the character which corresponds to the colour required. These can, as I said, be added as program eight shows.

Line 60 in program eight could also of course be added into the string, A$. Perhaps you might like to try to do this as an exercise. Program eight shows something else about the INPUT statement. It demonstrates that all the controls which are used for printing (including INK, PAPER and FLASH) can also be used to modify the INPUT statement, thus adding a considerable flexibility to the effects you can demand. The addition control characters are explained in the manual where there is a table giving a complete description of the various effects available from the top row of the keyboard.

If you want to see how effective the colour can be, even from a simple program, enter and run program nine. If the beeps drive you mad, delete lines 90 and 100. If you want the picutre to build up more quickly, change the 7 at the end of line 40 into a 6, so that white blobs are not printed.

When you've run this for a while, modify it to read as program nine b. You'll see this has BRIGHTned each blob, and added a random FLASH to each circuit of the program. BRIGHT and FLASH understand 1 as *on* (so FLASH 1 turns it on) and 0 as *off* (so FLASH 0 turns it off). FLASH and BRIGHT, like various other commands, do not INT a random number, but round it up or down to the nearest whole number (where the INT of a positive number is always the nearest whole number *below* the number plus fraction), so the effect of line 25 in program nine b is to turn the FLASH on for some loops of the program, and off for others. You can see this is so by changing the RND in line 25 to a 1, then running it for a while, then a 0 and running it for a while.

Finally, you may like to modify the program to become program nine c 'Greek alphabet soup', a name you will understand once you've seen the program running. This final version recaps many of the points we've discussed in this article.

```
  5 REM Program three
 10 LET a$=CHR$ 22+CHR$ 4+CHR$
7
 20 PRINT a$;"TEST"

  5 REM Program four
 10 LET a$=CHR$ 23+CHR$ 4
 20 PRINT a$;"xTEST"
 30 GO TO 20

 30 REM PROGRAM FIVE
 40 PRINT INK 4;"NORMAL ████"
 45 PRINT BRIGHT 1; INK 4;"BRIG
HT ████"
 50 PRINT INK 4; PAPER 2;"NORMA
L ████"
 55 PRINT BRIGHT 1; INK 4; PAPE
R 2;"BRIGHT ████"
 60 PRINT FLASH 1; INK 4;"FLASH
ING ████"
 65 PRINT BRIGHT 1; FLASH 1; IN
K 4;"BRIGHT ████"
 70 PRINT FLASH 1; PAPER 2; INK
 4;"FLASHING ████"
 75 PRINT BRIGHT 1; FLASH 1; PA
PER 2; INK 4;"BRIGHT ████"

 10 REM PROGRAM SIX
 60 PAPER RND*6
 70 CLS
 80 INK 9
100 FOR G=1 TO 32
110 PRINT AT RND*20,RND*30;CHR$
(65+INT (RND*26));
120 NEXT G
130 GO TO 60
140 PRINT AT RND*20,RND*30; OVE
R 1;CHR$ (65+INT (RND*26));

  5 REM PROGRAM SEVEN
 10 OVER 1
 20 FOR G=1 TO 16
 30 INPUT "ENTER A LETTER";A$
 40 INPUT "ENTER ANOTHER LETTER
";B$
 50 PRINT AT G,G;A$;CHR$ 8;B$
 60 NEXT G

 10 REM PROGRAM EIGHT
 20 INPUT PAPER 6; INK 1;"ENTER
 A COLOUR FOR INK";INK
 30 INPUT INK 2;"ENTER A COLOUR
FOR PAPER",PAPER
 40 INPUT FLASH 1; BRIGHT 1; IN
K 4; PAPER 2;"ENTER A WORD";A$
 50 LET A$=CHR$ 16+CHR$ INK+CHR
$ 17+CHR$ PAPER+A$
 60 PRINT AT 10,10;A$

  5 REM PROGRAM NINE
 10 PAPER 7: BORDER 0: CLS
 20 LET A=RND*10
 30 LET B=RND*16
 40 LET Z=RND*7
 50 PRINT AT A,B; INK Z;"█"
 60 PRINT AT 21-A,B; INK Z;"█"
 70 PRINT AT 21-A,31-B; INK Z;"
▒"
 80 PRINT AT A,31-B; INK Z;"█"
 90 IF RND>RND THEN GO TO 20
100 BEEP RND/30,RND*60-RND*60
110 GO TO 20

  5 REM PROGRAM NINE B
 10 PAPER 7: BORDER 0: CLS
 20 LET A=RND*10
 25 LET F=RND
 30 LET B=RND*16
 40 LET Z=RND*6
 50 PRINT AT A,B; FLASH F; BRIG
HT 1; INK Z;"█"
 60 PRINT AT 21-A,B; FLASH F; B
RIGHT 1; INK Z;"█"
 70 PRINT AT 21-A,31-B; FLASH F
; BRIGHT 1; INK Z;"█"
 80 PRINT AT A,31-B; FLASH F; B
RIGHT 1; INK Z;"█"
 90 IF RND>RND THEN GO TO 20
100 BEEP RND/30,RND*60-RND*60
110 GO TO 20

  5 REM PROGRAM NINE C
  7 REM GREEK ALPHABET SOUP
 10 PAPER 7: BORDER 0: CLS
 20 LET A=RND*10
 25 OVER 1
 30 LET B=RND*16
 40 LET Z=RND*7
 45 LET A$=CHR$ (65+RND*26)
 50 PRINT AT A,B; BRIGHT 1; INK
 Z;A$
 60 PRINT AT 21-A,B;; BRIGHT 1;
 INK Z;A$
 70 PRINT AT 21-A,31-B; BRIGHT
1; INK Z;A$
 80 PRINT AT A,31-B; BRIGHT 1;
INK Z;A$
 90 GO TO 20
```



# New ZX81 Software from Sinclair.




A whole new range of software for the Sinclair ZX81 Personal Computer is now available – direct from Sinclair. Produced by ICL and Psion, these really excellent cassettes cover games, education, and business/household management.

Some of the more elaborate programs can only be run on a ZX81 augmented by the ZX 16K RAM pack. (The description of each cassette makes it clear what hardware is required.) The RAM pack provides 16-times more memory in one complete module, and simply plugs into the rear of a ZX81. And the price has just been dramatically reduced to only £29.95.

The Sinclair ZX Printer offer full alphanumerics and highly-sophisticated graphics. A special feature is COPY which prints out exactly what is on the whole TV screen without the need for further instructions. So now you can print out your results for a permanent record. The ZX Printer plugs into the rear of your ZX81, and you can connect a RAM pack as well.

## Games

**Cassette G1: Super Programs 1 (ICL)**
*Hardware required* – ZX81.
*Price* – £4.95.
*Programs* – Invasion from Jupiter. Skittles. Magic Square. Doodle. Kim. Liquid Capacity.
*Description* – Five games programs plus easy conversion between pints/gallons and litres.

**Cassette G2: Super Programs 2 (ICL)**
*Hardware required* – ZX81.
*Price* – £4.95.
*Programs* – Rings around Saturn. Secret Code. Mindboggling. Silhouette. Memory Test. Metric conversion.
*Description* – Five games plus easy conversion between inches/feet/yards and centimetres/metres.

**Cassette G3: Super Programs 3 (ICL)**
*Hardware required* – ZX81.
*Price* – £4.95.
*Programs* – Train Race. Challenge. Secret Message. Mind that Meteor. Character Doodle. Currency Conversion.
*Description* – Fives games plus currency conversion at will – for example, dollars to pounds.

**Cassette G4: Super Programs 4 (ICL)**
*Hardware required* – ZX81.
*Price* – £4.95.
*Programs* – Down Under. Submarines. Doodling with Graphics. The Invisible Invader. Reaction. Petrol.
*Description* – Five games plus easy conversion between miles per gallon and European fuel consumption figures.

**Cassette G5: Super Programs 5 (ICL)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £4.95.
*Programs* – Martian Knock Out. Graffiti. Find the Mate. Labyrinth. Drop a Brick. Continental.
*Description* – Five games plus easy conversion between English and continental dress sizes.

**Cassette G6: Super Programs 6 (ICL)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £4.95.
*Programs* – Galactic Invasion, Journey into Danger. Create. Nine Hole Golf. Solitaire. Daylight Robbery.
*Description* – Six games making full use of the ZX81's moving graphics capability.

**Cassette G7: Super Programs 7 (ICL)**
*Hardware required* – ZX81.
*Price:* – £4.95.
*Programs* – Racetrack. Chase. NIM. Tower of Hanoi. Docking the Spaceship. Golf.
*Description* – Six games including the fascinating Tower of Hanoi problem.

**Cassette G8: Super Programs 8 (ICL)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £4.95.
*Programs* – Star Trail (plus blank tape on side 2).
*Description* – Can you, as Captain Church of the UK spaceship Endeavour, rid the galaxy of the Klingon menace?

**Cassette G9: Biorhythms (ICL)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £6.95.
*Programs* – What are Biorhythms? Your Biohythms.
*Description* – When will you be at your peak (and trough) physically, emotionally, and intellectually?

**Cassette G10: Backgammon (Psion)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £5.95.
*Programs* – Backgammon. Dice.
*Description* – A great program, using fast and efficient machine code, with graphics board, rolling dice, and doubling dice. The dice program can be used for any dice game.

**Cassette G11: Chess (Psion)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £6.95.
*Programs* – Chess, Chess Clock.
*Description* – Fast, efficient machine code, a graphic display of the board and pieces, plus six levels of ability, combine to make this one of the best chess programs available. The Chess Clock program can be used at any time.

**Cassette G12: Fantasy Games (Psion)**
*Hardware required* – ZX81 (or ZX80 with 8K BASIC ROM) + 16K RAM.
*Price* – £4.75.
*Programs* – Perilous Swamp. Sorcerer's Island.
*Description* – Perilous Swamp: rescue a beautiful princess from the evil wizard. Sorcerer's Island: you're marooned. To escape, you'll probably need the help of the Grand Sorcerer.

**Cassette G13: Space Raiders and Bomber (Psion)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £3.95.
*Programs* – Space Raiders. Bomber.
*Description* – Space Raiders is the ZX81 version of the popular pub game. Bomber: destroy a city before you hit a sky-scraper.

**Cassette G14: Flight Simulation (Psion)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £5.95.
*Program* – Flight Simulation (plus blank tape on side 2).
*Description* – Simulates a highly manoeuvrable light aircraft with full controls, instrumentation, a view through the cockpit window, and navigational aids. Happy landings!

## Education

**Cassette E1: Fun to Learn series – English Literature 1 (ICL)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £6.95.
*Programs* – Novelists. Authors.
*Description* – Who wrote 'Robinson Crusoe'? Which novelist do you associate with Father Brown?

**Cassette E2: Fun to Learn series – English Literature 2 (ICL)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £6.95.
*Programs* – Poets, Playwrights. Modern Authors.
*Description* – Who wrote 'Song of the Shirt'? Which playwright also played cricket for England?




**Cassette E3: Fun to Learn series – Geography 1 (ICL)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £6.95.
*Programs* – Towns in England and Wales. Countries and Capitals of Europe.
*Description* – The computer shows you a map and a list of towns. You locate the towns correctly. Or the computer challenges you to name a pinpointed location.

**Cassette E4: Fun to Learn series – History 1 (ICL)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £6.95.
*Programs* – Events in British History. British Monarchs.
*Description* – From 1066 to 1981, find out when important events occurred. Recognise monarchs in an identity parade.

**Cassette E5: Fun to Learn series – Mathematics 1 (ICL)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £6.95.
*Programs* – Addition/Subtraction. Multiplication/Division.
*Description* – Questions and answers on basic mathematics at different levels of difficulty.

**Cassette E6: Fun to Learn series – Music 1 (ICL)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £6.95.
*Programs* – Composers. Musicians.
*Description* – Which instrument does James Galway play? Who composed 'Peter Grimes'?

**Cassette E7: Fun to Learn series – Inventions 1 (ICL)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £6.95.
*Programs* – Inventions before 1850. Inventions since 1850.
*Description* – Who invented television? What was the 'dangerous Lucifer'?

**Cassette E8: Fun to Learn series – Spelling 1 (ICL)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £6.95.
*Programs* – Series A1-A15. Series B1-B15.
*Description* – Listen to the word spoken on your tape recorder, then spell it out on your ZX81. 300 words in total suitable for 6-11 year olds.

## Business/household

**Cassette B1: The Collector's Pack (ICL)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £9.95.
*Program* – Collector's Pack, plus blank tape or side 2 for program/data storage.
*Description* – This comprehensive program should allow collectors (of stamps, coins etc.) to hold up to 400 records of up to 6 different items on one cassette. Keep your records up to date and sorted into order.

**Cassette B2: The Club Record Controller (ICL)**
*Hardware required* – ZX81 + 16K RAM.
*Price* – £9.95.
*Program* – Club Record Controller plus blank tape on side 2 for program/data storage.
*Description* – Enables clubs to hold records of up to 100 members on one cassette. Allows for names, addresses, 'phone numbers plus five lots of additional information – eg type of membership.

**Cassette B3: VU**
*Hardware requirec*
*Price* – £7.95.
*Program* – VU-CALC.
*Description* – Turns yc immensely powerful ar. VU-CALC constructs, ge calculates large tables fo. such as financial analysis, sheets, and projections. Co full instructions.

**Cassette B4: VU-FILE (Psion)**
*Hardware required* – ZX81 + 16K ..M.
*Price* – £7.95.
*Programs* – VU-FILE. Examples.
*Description* – A general-purpose information storage and retrieval program with emphasis on user-friendliness and visual display. Use it to catalogue your collection, maintain records or club memberships, keep track of your accounts, or as a telephone directory.

## How to order

Simply use the FREEPOST order form below and either enclose a cheque or give us your credit card number. Credit card holders can order by phone – simply call Camberley (0276) 66104 or 21282 during office hours. Either way, please allow up to 28 days for delivery, and there's a 14-day money-back option, of course.

# sinclair ZX81 SOFTWARE

**Sinclair Research Ltd,**
Stanhope Road, Camberley, Surrey, GU15 3PS.
Tel: Camberley (0276) 66104 & 21282.

To: Sinclair Research, FREEPOST, Camberley, Surrey, GU15 3BR.
Please send me the items I have indicated below.

| Qty | Cassette | Code | Item price | Total |
|---|---|---|---|---|
| | G1: Super Programs 1 | 30 | £4.95 | |
| | G2: Super Programs 2 | 31 | £4.95 | |
| | G3: Super Programs 3 | 32 | £4.95 | |
| | G4: Super Programs 4 | 33 | £4.95 | |
| | G5: Super Programs 5 | 34 | £4.95 | |
| | G6: Super Programs 6 | 35 | £4.95 | |
| | G7: Super Programs 7 | 36 | £4.95 | |
| | G8: Super Programs 8 | 37 | £4.95 | |
| | G9: Biorhythms | 38 | £6.95 | |
| | G10: Backgammon | 39 | £5.95 | |
| | G11: Chess | 40 | £6.95 | |
| | G12: Fantasy Games | 41 | £4.75 | |
| | G13: Space Raiders & Bomber | 42 | £3.95 | |
| | G14: Flight Simulation | 43 | £5.95 | |
| | E1: English Literature 1 | 44 | £6.95 | |

| Qty | Cassette | Code | Item price | Total |
|---|---|---|---|---|
| | E2: English Literature 2 | 45 | £6.95 | |
| | E3: Geography 1 | 46 | £6.95 | |
| | E4: History 1 | 47 | £6.95 | |
| | E5: Mathematics 1 | 48 | £6.95 | |
| | E6: Music 1 | 49 | £6.95 | |
| | E7: Inventions 1 | 50 | £6.95 | |
| | E8: Spelling 1 | 51 | £6.95 | |
| | B1: Collector's Pack | 52 | £9.95 | |
| | B2: Club Record Controller | 53 | £9.95 | |
| | B3: VU-CALC | 54 | £7.95 | |
| | B4: VU-FILE | 55 | £7.95 | |
| | ZX 16K RAM pack | 18 | £29.95 | |
| | ZX Printer | 27 | £59.95 | |
| | Post & packing – only if ordering hardware | | £2.95 | |

TOTAL £ ____

I enclose a cheque/postal order to Sinclair Research Ltd for £ ____

Please charge my *Access/Barclaycard/Trustcard no.
**Please delete as applicable.*

Mr/Mrs/Miss

Address

NSA14

# MEMOTECH Explores the Excellence of your ZX81




# Memotech's Memopak Range

All five of the currently available Memopaks are housed in elegant black anodised aluminium cases, and are styled to fit wobble-free onto the back of the ZX8l, allowing more add-ons (from Memotech or Sinclair) to be connected.

**£68.70 plus VAT**

## MEMOPAK 64K MEMORY EXTENSION

The 64K Memopak extends the memory of the ZX81 by 56K, and with the ZX81 gives 64K, which is neither switched nor paged and is directly addressable. The unit is user transparent and accepts commands such as 10 DIM A(9000).

Breakdown of memory areas...0-8K-Sinclair ROM. 8-16K-This area can be used to hold machine code for communication between programmes or peripherals. 16-64K-A straight 48K for normal Basic use.

**£43.43 32K / £26.00 16K plus VAT**

## MEMOPAK 32K and 16K MEMORY EXTENSIONS

These two packs extend and complete the Memotech RAM range (for the time being!) A notable feature of the 32K pack is that it will run in tandem with the Sinclair 16K memory extension to give 48K RAM total.

**£52.00 plus VAT**

## MEMOPAK HIGH RES GRAPHICS PACK

HRG Main Features — • Fully programmable Hi-Res (192 × 248 pixels) • Video page is both memory and bit mapped and can be located anywhere in RAM. • Number of Video pages is limited only by RAM size (each takes about 6.5K RAM) • Instant inverse video on/off gives flashing characters • Video pages can be superimposed • Video page access is similar to Basic plot/unplot commands • Contains 2K EPROM monitor with full range of graphics subroutines controlled by machine code or USR function

**£34.70 plus VAT**

## MEMOPAK CENTRONICS TYPE PARALLEL PRINTER INTERFACE

Main Features — • Interfaces ZX8l and parallel printers of the Centronics type • Enables use of a range of dot matrix and daisy wheel printers with ZX8l • Compatible with ZX8l Basic, prints from LLIST, LPRINT and COPY • Contains firmware to convert ZX8l characters to ASCII code • Gives lower-case characters from ZX8l inverse character set

## Coming Soon...




A complete range of ZX81 plug-in peripherals

Digitising Tablet RS232 Interface

We regret we are as yet unable to accept orders or enquiries concerning the above products, but we'll let you know as soon as they become available.

Access/Barclaycard Holders Please Tel. Oxford (0865) 722102 (24 Hrs)

© R. Branton & G.A.C. Boyd 1982

Please make cheques payable to MEMOTECH Ltd.

Please Debit my Access/Barclaycard* account number

*Please delete whichever does not apply

| Please send me | Price | No | Total |
|---|---|---|---|
| 64K RAM £68.70 + £10.30 VAT | £79.00 | | |
| 32K RAM £43.43 + £6.52 VAT | £49.95 | | |
| 16K RAM £26.00 + £3.90 VAT | £29.90 | | |
| HRG £52.00 + £7.80 VAT | £59.80 | | |
| CENTRONICS I F £34.70 + £5.20 VAT | £39.90 | | |
| Packaging & Postage £2.00 per unit | | | |
| TOTAL ENC | | | |

SIGNATURE ____________ DATE ____________

NAME ____________ ADDRESS ____________

____________ TELEPHONE ____________

We want to be sure you are satisfied with your Memopak – so we offer a 14-day money back Guarantee on all our products.

Memotech Limited, 3 Collins Street, Oxford OX4 1XL, England Tel: Oxford (0865) 722102 Telex: 837220 Orchid G

# MACHINE SPECIFICATIONS

## ZX80

Dimensions
Width 174mm (6.85 in)
Depth 218mm (8.58 in)
Height 38 mm (1. 5 in )
Weight 300g (10.5oz)

Microprocessor/Memory
Z80A 3.25 MHz clock
ROM: 4K bytes containing BASIC
RAM: 1K bytes internal, externally expandable to 16K bytes.

Display
Requires an ordinary domestic black and white colour TV. The lead supplied connects between the ZX80 and your TV's aerial socket. The display organisation is 24 lines of 32 characters per line showing black characters on a white screen. The ZX80 does not connect to a printer.
Programming
Programs can be entered on the keyboard or loaded from cassette. The ZX80 has automatic "wrap round" so lines of program can be any length but not multi-statement lines.
Syntax check
The syntax of the entered line is checked character by character. A syntax error cursor marks the first place the syntax breaks down if there is an error. Once any errors have been edited out the syntax error cursor disappears. Only syntax error-free lines of code are accepted by the ZX80.
Graphics
Total of 22 graphics symbols giving 48 x 64 pixels resolution consisting of 10 symbols plus space and inverses. Includes symbols for drawing bar charts. Under control of your BASIC program any character can be printed in reverse field.
Editing
The line edit allows you to edit any line of program or input including statement numbers. The edit and cursor control keys are EDIT, RUBOUT, HOME.
Arithmetic
Arithmetic operators +,–,x, ÷ exponentiate. Relational operators <, >, = , yielding 0 or – 1. Logical operators AND OR NOT yielding boolean result. Relational operators also apply to strings. ZX80 BASIC uses 16 bit two's complement arithmetic ( ± 32767 ).
Variables
Numeric variable names may be any length, must begin with a letter and consist of alphanumerics. Every character in the name is compared thus an infinity of unique names is available.

String variables may be assigned to or from, shortened but not concatenated. String variable names are A$ – Z$. Strings do not require a dimension statement and can be any length.

Arrays have a maximum dimension of 255 (256 elements) each. Array names consist of a single letter A–Z.
Control variable names in FOR. . . NEXT loops consist of a single letter A–Z.

Expression evaluator
The full expression evaluator is called whenever a constant or variable is encountered during program execution. This allows you to use expressions in place of constants especially useful in GOTOs, GOSUBs, FOR. . . NEXT etc.

Immediate mode
The ZX80 will function in the "calculator mode" by immediately executing a statement if it is not preceded with a line number.
Cassette interface
Works with most domestic cassette recorders. The transfer rate is 250 baud using a unique tape-recording format. Other systems are not compatible with the ZX80's. The ZX80 also SAVEs the variables as well as the program on cassette. Therefore you can save the data for updating next time the program is executed. The ZX80 does not support separate data files. The lead supplied with the ZX80 is fitted with 3.5mm jack plugs.
Expansion bus
At the rear has 8 data, 16 address, 13 control lines from the processor and Ov, 5v, 9-11v, Ø and internal memory control line. These signals enable you to interface the ZX80 to your own electronics, PIO, CTC, SIO if you want I/O ports etc.
Power supply
The ZX80 requires approximately 400mA from 7–11v DC. It has its own internal 5v regulator.
TV standard
The ZX80 is designed to work with UHF TVs (channel 36)and is the version required for use in the United Kingdom. The ZX80 USA is designed to work with a VHF TV(American channel 2. European channel 3) and is the version required for the American TV system, also for countries without UHF.

## ZX81

Dimensions
Width 167mm (6.32 in)
Depth 175mm (6.80 in)
Height 40 mm (1.57 in)
Weight 350 gms (12.15 oz)

Microprocessor/Memory
Z80A 3.25 MHz clock
ROM: Containing 8K BASIC interpreter
RAM: 1K bytes internal, externally expandable to 16K bytes.

Keyboard
40 key touch-sensitive membrane. Using function mode and single press key-word system, this gives the equivalent of 91 keys and also graphics mode allows an additional 20 graphical and 54 inverse video characters to be entered directly.
Display
Requires an ordinary domestic black and white or colour TV. The aerial lead supplied connects the ZX81 to the TV aerial socket. The display is organised as 24 lines of 32 characters with black characters on a white background.
Two mode speeds
The ZX81 can operate in two software-selectable modes - FAST and NORMAL. FAST is ideal for really high-speed computing. In NORMAL mode however the ZX81 allows continuously moving, flicker-free animated displays.
Printer

The 8K ROM will permit instructions (LPRINT, LLIST and COPY) to drive the Sinclair ZX Printer.
Programming
Programs can be entered via the keyboard or loaded from cassette. Programs and data can be saved onto cassette so that they

are not lost when the ZX81 is turned off.

Syntax check

The syntax of a line of program is checked on entry. A syntax error cursor marks the first place the syntax breaks down if there is an error. The syntax error cursor disappears when errors have been corrected. Only lines free from syntax errors will be entered into the program.

Graphics

Apart from the 20 graphics characters, space and its inverse, the display may also be divided into 64 x 44 pixels, each of which may be 'blacked' in or 'whited' out under program control.

Editing

A line editor allows you to edit any line of program or input, including program line numbers. Lines may be deleted, increased or decreased in size.

Arithmetic

Arithmetic operators +, –, x, ÷, exponentiate. Relational operators =, < >, >, <,< =,>=, may compare string and arithmetic variables to yeild 0 (False) or 1(True). Logical operators AND, OR, NOT yield boolean results.

Floating-point numbers

Numbers are stored in 5 bytes in floating-point binary form giving a range of $\pm 3 \times 10^{-39}$ to $\pm 7 \times 10^{38}$ accurate to 9½ decimal digits.

Scientific functions

Natural logs/antilogs; SIN, COS, TAN and their inverses;SQR; $e^x$.

Variables

| | |
|---|---|
| Numerical: | any letter followed by alphanumerics |
| String: | A$ to Z$ |
| FOR-NEXT loops: | A–Z (loops may be nested to any depth. |
| Numerical arrays: | A–Z |
| String arrays: | A$ to Z$ |

Arrays

Arrays may be multi-dimensional with subscripts starting at 1.

Expression evaluator

The full expression evaluator is called whenever an expression, constant or variable is encountered during program execution. This powerful feature allows use of expressions in place of constants and is especially useful in GOTO, GOSUB etc.

Command mode

The ZX81 will execute statements immediately, enabling it to perform like a calculator.

Cassette interface

Works using domestic cassette recorders. The transfer rate is 250 baud and uses a unique recording format not compatible with other systems. The ZX81 will save the data as well as the program to avoid the need to re-enter the data when the program is next loaded.,

ZX81 will search through a tape for the required program). The cassette leads supplied have 3.5 mm jack plugs.

Expansion port

At the rear, this has the full data, address and control buses from the Z80A CPU as well as OV, +5V, +9V,Ø and the memory select lines. These signals enable you to interface the ZX81 to the Sinclair 16K RAM pack and ZX printer.

Power supply

The ZX81 requires approximately 420mA at 7–11V DC. It has its own internal 5V regulator. The ready assembled ZX81 comes complete with a power supply. The ZX81 kit does not include a power supply.

TV standard

The ZX81 is designed to work with UHF TVs (channel 36) 625 lines.

# ZX SPECTRUM

**Dimensions**

Width 233 mm
Depth 144 mm
Height 30 mm

**CPU/Memory**

Z80A microprocessor running at 3.5 MHz. 16K-byte ROM containing BASIC interpreter and operating system.
16K-byte RAM (plus optional 32K-byte RAM on internal expansion board) or 48K-byte RAM.

**Keyboard**

40-key keyboard with upper and lower case with capitals lock feature. All BASIC words obtained by single keys, plus 16 graphics characters, 22 colour control codes and 21 user-definable graphics characters. All keys have auto repeat.

**Display**

Memory-mapped display of 256 pixels x 192 pixels; plus one attributes byte per character square, defining one of eight foreground colours, one of eight background colours, normal or extra brightness and flashing or steady. Screen border colour also settable to one of eight colours. Will drive a PAL UHF colour TV set, or black and white set (which will give a scale of grey), on channel 36.

**Sound**

Internal loudspeaker can be operated over more than 10 octaves (actually 130 semitones) via basic BEEP command. Jack sockets at the rear of computer allow connections to external amplifier/speaker.

**Graphics**

Point, line, circle and arc drawing commands in high-resolution graphics.
16 pre-defined graphics characters plus 21 user-definable graphics characters. Also functions to yield character at a given position, attribute at a given position (colours, brightness and flash) and whether a given pixel is set. Text may be written on the screen on 24 lines of 32 characters. Text and graphics may be freely mixed.

**Colours**

Foreground and background colours, brightness and flashing are set by BASIC INK, PAPER, BRIGHT and FLASH commands. OVER may also be set, which performs an exclusive — or operation to overwrite any printing or plotting that is already on the screen. INVERSE will give inverse video printing. These six commands may be set globally to cover all further PRINT, PLOT, DRAW or CIRCLE commands, or locally within these commands to cover only the results of that command. They may also be set locally to cover text printed by an INPUT statement. Colour-control codes, which may be accessed from the keyboard, may be inserted into text or program listing, and when displayed will override the globally set colours until another control code is encountered. Brightness and flashing codes may be inserted into program or text, similarly. Colour-control codes in a program listing have no effect on its execution. Border colour is set by a BORDER command. The eight colours available are black, blue, red, magneta, green, cyan, yellow and white. All eight colours may be present on the screen at once, with some areas flashing and others steady, and any area may be highlighted extra bright.

**Screen**

The screen is divided into two sections. The top section — normally the first 22 lines — displays the program listing or the results of program or command execution. The bottom section — normally the last 2 lines — shows the command or program line currently being entered, or the program line currently being edited. It also shows the report messages. Full editing facilities of cursor left, cursor right, insert and delete (with auto-repeat facility) are available over this line. The bottom section will expand to accept a current line of up to 22 lines.




### Mathematical Operations And Functions

Arithmetic operations of +, −, ×, ÷, and raise to a power. Mathematical functions of sine, cosine, tangent and their inverses; natural logs and exponentials; sign function, absolute value function, and integer function; square root function, random number generation, and pi.
Numbers are stored as five bytes of floating point binary — giving a range of $+3 \times 10^{-39}$ to $+7 \times 10^{38}$ accurate to 9½ decimal digits.
Binary numbers may be entered directly with the BIN function. =, >, <, >=, <= and <> may be used to compare string or arithmetic values or variables to yield 0 (false) or 1 (true). Logical operators AND, OR and NOT yield boolean results but will accept 0 (false) and any number (true).
User-definable functions are defined using DEF FN, and called using FN. They may take up to 26 numeric and 26 string arguments, and may yield string or numeric results.
There is a full DATA mechanism, using the commands READ, DATA and RESTORE.
A real-time clock is obtainable.

### String Operations And Functions

Strings can be concatenated with +. String variables or values may be compared with =, >, <, >=, <=, <> to give boolean results. String functions are VAL, VAL$, STR$ and LEN. CHR$ and CODE convert numbers to characters and vice versa, using the ASCII code. A string slicing mechanism exists, using the form a$ (x TO y).

### Variable Names

Numeric — any string starting with a letter (upper and lower case are not distinguished between, and spaces are ignored).
String — A$ to Z$.
FOR-NEXT loops — A-Z.
Numeric arrays — A-Z.
String arrays — A$ to Z$.
Simple variables and arrays with the same name are allowed and distinguished between.

### Arrays

Arrays may be multi-dimensional, with subscripts starting at 1. String arrays, technically character arrays, may have their last subscript omitted, yielding a string.

### Expression Evaluator

A full expression evaluator is called during program execution whenever an expression, constant or variable is encountered. This allows the use of expressions as arguments to GOTO, GOSUB, etc.
It also operates on commands allowing the ZX Spectrum to operate as a calculator.

### Cassette Interface

A tone leader is recorded before the information to overcome the automatic recording level fluctuations of some tape recorders, and a Schmitt trigger is used to remove noise on playback.
All saved information is started with a header containing information as to its type, title, length and address information. Program, screens, blocks of memory, string and character arrays may all be saved separately.
Programs, blocks of memory and arrays may be verified after saving.
Programs and arrays may be merged from tape to combine them with the existing contents of memory. Where two line numbers or variables names coincide, the old one is overwritten.
Programs may be saved with a line number, where execution will start immediately on loading.
The cassette interface runs at 1500 baud, through two 3.5 mm jack plugs.

### Expansion Port

This has the full data, address and control busses from the Z80A, and is used to interface to the ZX Printer, the RS232 and NET interfaces and the ZX Microdrives. IN and OUT commands give the I/O port equivalents of PEEK and POKE.

### ZX81 Compatibility

ZX81 BASIC is essentially a subset of ZX Spectrum [illegible]C. The differences are as follows.
FAST and SLOW: the ZX Spectrum operates at th[illegible] of the ZX81 in FAST mode with the steady display of SL[illegible], and does not include these commands.
SCROLL: the ZX Spectrum scrolls automatical[illegible] the operator "scroll?" every time a screen is filled.
UNPLOT: the ZX Spectrum can unplot a pixel usi[illegible]VER, and thus achieves unplot.
Character set: the ZX Spectrum uses the ASCII c[illegible], as opposed to the ZX81 non-standard set.

# ADVERTISEMENT INDEX

# FULLER FD SYSTEM £39.95

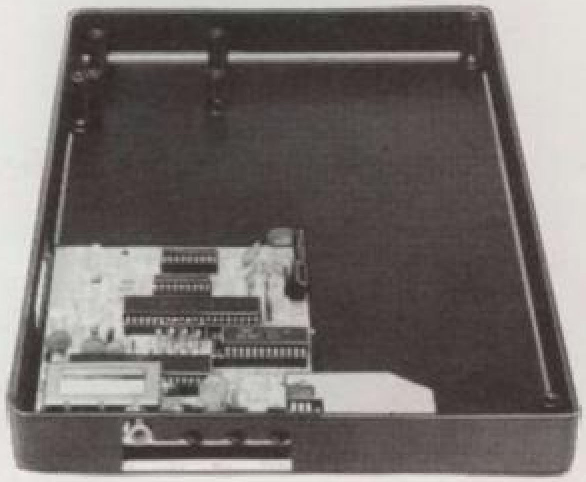
*Professional Keyboard & Case for Sinclair ZX81 & ZX Spectrum*

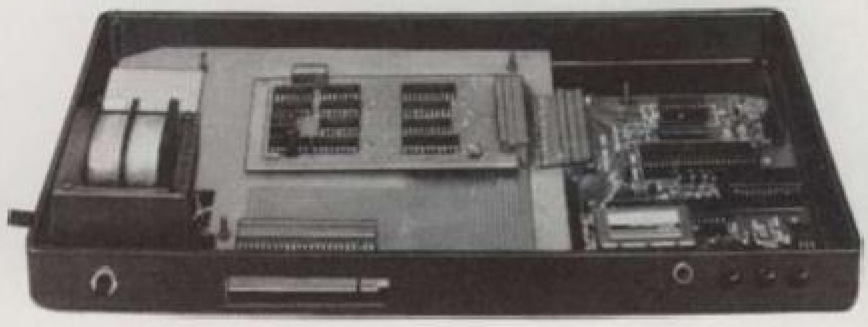


**The ZX81 fits inside**
The tough ABS injection moulded plastic case measures 8" x 14" x 2½" and hooks up to your ZX printed circuit board in minutes. No technical know how or soldering is required.

**The ZX16K Memory Module** will fix inside the case, using the new Adaptor Board at £9.75 or the Motherboard.
By removing the ZX PSU from its case this can also be fixed inside. We will carry out the installation work free of charge if required.

**KEYBOARD LAYOUT:**
All the Sinclair ZX81 keys are duplicated on our layout, with extra shift and new line keys. The professional momentary action key switches have a guaranteed life of 10⁶ operations. The unit is fully built tested and comes complete with a money back guarantee.





**INSTALLATION**
Simply unscrew the ZX printed circuit board from its case and screw it into the FD Case.

**MOTHERBOARD:**
We also manufacture a mother board which allows expansion to the ZX memory and 1/0 facilities WITHIN the case, as well as our power supply unit and reset switch.

AD Code

| | | Price † | Qty | Amount |
|---|---|---|---|---|
| Fuller | FD System 42 Keyboard & case | 39.95 | | |
| | FD System Motherboard | 15.95 | | |
| | FD 16K. Memory Module | 29.95 | | |
| | FD 64K. Memory Module | 78.95 | | |
| | FD PSU 9 Volts at 2 amp. | 12.95 | | |
| | FD *Shipping and Handling* | 2.50 | | |

Keyboard Only Available!
Built £24.95
Kit £18.95 (+P.P. 80p)

Mail to **FULLER MICRO SYSTEMS,**
The ZX Centre, Sweeting Street, Liverpool 2. England, U.K.
SAE for more details — Enquiries: Tel. 051-236 6109

Name ..........................................

Address ..........................................

City/State/Zip ..........................................


© RICH SHENFIELD 82